

## **KOREAN PATENT ABSTRACTS**

(11)Publication number: **10-2005-0025220 A**

(43)Date of publication of application: **14.03.2005**

(51)Int. Cl.

**G06F 3 / 14**

(21)Application number: **10-2003-0101713**

(22)Date of filing: **31.12.2003**

(30)Priority: **60/500,669(05. 09. 2003) US**  
**10/743,476(23. 12. 2003) US**

(71)Applicant: **SAMSUNG ELECTRONICS CO., LTD.**

(72)Inventor: **BEN YAIR RAN**  
**EISENBERG YARIV**  
**LEE, JONG GOO**  
**LINDER NATAN**  
**TOLEDANO EYAL**

---

### **(54) PROACTIVE USER INTERFACE FOR MOBILE INFORMATION DEVICE**

(57) Abstract:

PURPOSE: A proactive user interface for a mobile information device is provided to offer a flexible and actually user-friendly interface, and improved user experience and interaction with an operation device, and offer improved emotional experience with the operation device by forming the user interface in an organism or avatar form interacting with a user.

CONSTITUTION: The user interface performs communication between the user and an OS(Operating System). A learning module detects at least one pattern of the interaction between the user interface and the user and changes at least one function of the user interface to a proactive type according to the detected pattern.

(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(51) . Int. Cl.<sup>7</sup>  
G06F 3/14

(11) 공개번호 10-2005-0025220  
(43) 공개일자 2005년03월14일

(21) 출원번호 10-2003-0101713  
(22) 출원일자 2003년12월31일

(30) 우선권주장 60/500,669 2003년09월05일 미국(US)  
10/743,476 2003년12월23일 미국(US)

(71) 출원인 삼성전자주식회사  
경기도 수원시 영통구 매탄동 416

(72) 발명자 이종구  
서울특별시서대문구홍제동360번지65호(23/1)

톨레대노에이얼  
이스라엘, 28000, 카르예트아타, 자비헤르스트리트, 52

란다나탄  
이스라엘, 90820, 오타즈일리트, 하타포우치스트리트, 21

에이슨버그야리브  
이스라엘, 텔아비브 18버그슨스트리트, #5아파트

벤-야이르란  
이스라엘, 90860, 헤리예우다임.피., 모스하브쇼울쉬, #63

## (74) 대리인 이건주

### 심사 청구 : 없음

#### (54) 사전 행동적 사용자 인터페이스

요약

어떠한 타입의 연산장치에도 임의로 설치될 수 있는(혹은 그 장치를 제어하거나 그 장치와 연관될 수 있는) 사전 행동적 사용자 인터페이스이다. 이 사전행동적 사용자 인터페이스는 사용자의 행동에 따라, 연산장치가 선택할 수 있는 미리 프로그램된 여러 가지 패턴들 및/또는 특정한 사용자와의 이전 경험에 기초하여, 사용자에게 제안들을 능동적으로 행한다. 예를 들어 메뉴 또는 그 일부를 변경하고; 표시할 다른 메뉴를 제공하고; 및/또는 터치스크린 기능을 변경하는 등 표시부의 적어도 일부의 모습을 변경함으로써 그 제안들을 임의로 행할 수 있다. 이러한 제안들은 음성에 의해 임의로 행해질 수 있다.

대표도

도 2

색인어

프로엑티브, 인공지능, 가상세계, 인터페이스, 아바타, 휴대폰, 쌍방향

명세서

## 도면의 간단한 설명

도 1은 본 발명에 따른 학습모듈의 예를 나타낸 개략 블록도,

도 2는 사전행동적 사용자 인터페이스를 이용하기 위한 본 발명에 따른 시스템의 예를 나타낸 개략 블록도,

도 3은 본 발명에 따른 사전행동적 사용자 인터페이스 시스템의 구현 예를 나타낸 도면,

도 4는 본 발명에 따른 적응형 시스템의 구현 예를 나타낸 개략 블록도,

도 5a 및 도 5b는 각각 본 발명에 따른 애플리케이션 관리 시스템의 예를 나타낸 개략 블록도 및 시퀀스 다이어그램,

도 6a 및 도 6b는 각각 이동 정보 장치의 운영체제를 통하여 하나 이상의 행위를 수행하기 위해 본 발명에 따른 적응형 시스템에게 필요한 기반구조(infrastructure)의 예를 나타낸 도면과, 이것에 대한 본 발명에 따른 시퀀스 다이어그램의 예를 나타낸 도면,

도 7a 내지 도 7c는 전형적인 이벤트들을 나타낸 것으로, 그 이벤트들이 (이동 정보 장치의 운영체제를 통해) 이동 정보 장치와 본 발명의 시스템 사이의 상호작용들에 의해 어떻게 처리되는지를 나타낸 도면,

도 8a는 지능형 에이전트의 구성 예를 나타낸 것이고, 도 8b는 그 지능형 에이전트의 동작에 대한 시퀀스 다이어그램의 예를 나타낸 도면,

도 9a 및 도 9b는 본 발명에 따른 행위를 선택하기 위한 예로서 2가지 방법을 나타낸 도면,

도 10은 본 발명에 따른 행위 수행(action execution)의 예를 나타낸 시퀀스 다이어그램,

도 11a 내지 도 11c는 본 발명에 따른 전형적인 감정형 시스템의 구현 예를 기술하기 위한 다이어그램을 특징으로 하는 도면,

도 12는 본 발명에 따른 텍스트 형식의 통신을 위한 전형적인 시퀀스 다이어그램,

도 13a 및 도 13b는 각각 본 발명에 따른 통화 처리(telephone call handling)를 위한 전형적인 클래스 다이어그램 및 전형적인 시퀀스 다이어그램,

도 14a는 본 발명에 따른 SMS 메시지 처리 클래스의 예시적이고 비한정적인 예들을 나타낸 도면이고, 도 14b는 그 것에 대한 시퀀스 다이어그램,

도 15는 본 발명에 따른 전형적인 메뉴 처리 클래스 다이어그램,

도 16은 본 발명에 따른 전형적인 게임 클래스 다이어그램,

도 17a는 전형적인 교육 기계 클래스 다이어그램이고, 도 17b는 본 발명에 따른 전형적인 교육 기계 시퀀스 다이어그램,

도 18a 내지 도 18c는 각각 본 발명에 따른 전형적인 전화 클래스 다이어그램과, 전형적인 돌연변이 시퀀스 다이어그램과, 전형적인 혼성 시퀀스 다이어그램,

도 19는 2개의 이동 정보 장치 상의 지능형 에이전트들 사이의 전형적인 혼성화 과정(hybridization sequence)을 나타낸 도면,

도 20 내지 도 26은 본 발명에 따른 아바타 혹은 생물의 전형적인 화면출력화상들(screenshots)을 나타낸 도면,

도 27은 본 발명에 따른 전형적인 지능형 에이전트 시스템을 나타낸 개략 블록도,

도 28은 도 27의 시스템을 더욱 자세히 나타낸 도면,

도 29는 본 발명에 다른 행위 선택 시스템의 전형적인 구현예를 나타낸 개략 블록도,

도 30a 및 도 30b는 이동 정보 장치의 스크린 상에 표시된 본 발명에 따른 아바타의 몇 가지 화면출력화상들을 나타낸 도면.

**발명의 상세한 설명**

**발명의 목적**

## 발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 사전행동적 사용자 인터페이스(proactive user interface)와, 그 시스템 및 방법에 관한 것으로, 특히 이동 정보 장치와 함께 사용하기 위한 사용자 인터페이스와, 그 시스템 및 방법에 관한 것이다.

최근, 이동 및 휴대 무선 장치의 사용이 급격히 증가하고 있다. 변화하는 기능들, 내부 자원들 및 성능들을 가지고 있는 많은 장치들이 있으며, 이러한 장치에는, 이동 전화, PDA(Personal Digital Assistants), 의료용 및 실험실용 기구, 스마트카드, 및 셋탑박스 등이 포함되지만, 그것들에 한정되는 것은 아니다. 그러한 장치는 모두 이동 정보 장치이다. 이들 장치는 이미 알려져 있는 범용 계산 기계보다는 특수 목적용의 기능 제한형 장치가 되는 경향이 있다. 이들 중 많은 장치는 인터넷에 연결되어 여러 가지 용도로 이용되고 있다.

그러한 이동 정보 장치로는 예를 들어 휴대폰(cellular phone)이 있다. 휴대폰은 급속도로 보편화되어, 휴대폰의 사용이 심지어는 기존의 PSTN(Public Switched Telephony Network) 전화기 혹은 "지상선로(land line)" 전화기의 사용을 능가하고 있다. 휴대폰은 그 자체로 더욱 정교해지고 있으며, 실제로는, 사실상의 내장형 운영체제를 갖춘 연산 장치라고 할 수 있다.

휴대폰이 더욱 정교해짐에 따라, 휴대폰에 의해 제공되는 기능의 범위도 잠재적으로 더욱 확대되고 있다. 하지만, 현재로는, 휴대폰의 기능들은 일반(지상선로) 전화기에 이미 있는 기능들을 확장하는 것 및/또는 PDA의 특정 기능들을 휴대폰의 기능들과 통합하는 것과 관련되는 것이 일반적이다. 이와 유사하게, 휴대폰과 함께 제공되는 사용자 인터페이스도 정교하지 않은 것이며, 몇 가지 단순한 메뉴들을 스크롤하기 위한 키패드를 특징적 요소로서 갖추는 것이 일반적이다. 벨 소리 및 그 이외의 휴대폰 액세서리들을 위해 상당한 비용을 지출하고 있는 고객들 입장에서는 고객 맞춤화(customization)를 분명 원하고 있으나, 고객 맞춤화는 여전히 휴대폰의 극히 적은 개수의 기능들에 한정되고 있다. 또한, 휴대폰은, 현재, 이동 정보 장치의 사용을 향상시키는데 필요한 주문 제작형 기능(custom/tailored functionalities)과, 예를 들어 장치 사용자 인터페이스 등의 자동 개별 맞춤화(automatic personalization), 및/또는 사용자의 행동에 따라 반응하는 능력이 부족하다.

그러나, 이와 같은 정교함의 부족한 것은 개인용 (데스크탑 혹은 랩탑) 컴퓨터용 및 그 이외의 연산장치용 사용자 인터페이스들에서도 마찬가지이다. 대개 이들 연산장치는 아주 단순한 방법으로만 고객 맞춤화될 수 있다. 또한, 그러한 고객 맞춤화는 사용자에 의해 수행되어야 하는데, 사용자들은 컴퓨터기능들을 이해하지 못하거나 그러한 고객 맞춤화 작업의 수행을 불편하게 느낄 수 있다. 현재, 연장 장치들은 사용자의 행동(동작) 패턴을 학습하여 그에 따라 그들 자신의 동작을, 사용자 인터페이스용 적응형 시스템(adaptive system)처럼, 조정하는 것이 불가능하다. 사용자가 컴퓨터를 수동으로 조정할 수 없다면, 컴퓨터가 사용자에 맞도록 그 동작을 조정하기보다는, 사용자가 자신의 동작을 컴퓨터에 맞도록 조정해야 한다.

전문화된 실험실용 기능들만을 위한 것이지만 학습할 수 있는 소프트웨어가 개발된 바 있다. 예를 들어, "인공지능(AI: Artificial Intelligence)"라는 소프트웨어가 개발되었다. 인공지능이란 용어에 대한 여러 가지 정의가 있는데, 그 중에 하나로서, "인공지능이란 인지, 추론 및 행동할 수 있도록 하는 연산들의 연구"라고 정의된다(Prentice Hall, Pearson Education Inc.의, 2003년, Stuart Russell, Peter Norvig의 "Artificial Intelligence: A Modern Approach (second edition)"에서 인용함). 인공지능 소프트웨어는 인지(perception) 등의 여러 가지 다른 개념들을 결합한 것으로, 이는 추론하고 행동하기 위해서 인공지능 소프트웨어를 필요로 하는 세계(world)에 인터페이스를 제공하게 된다. 그 예로는, 자연언어 처리(natural language processing)(자연언어의 문헌 내용 및 문맥을 전달(communication)하고 및 이해함)과; 컴퓨터 비전(화상원(imagery source)으로부터 객체(objects)를 인지함)과; 센서시스템(감각 데이터(sensory data)를 분석하여 인지된 객체들의 특징들(features) 및 객체들을 인지함) 등이 포함되나, 그것들에 한정되는 것은 아니다.

또 다른 중요한 개념으로는 지식베이스(knowledge base)라는 것이 있다. 지식 표현(knowledge representation)은 지식의 추출 및 저장을 표현하는 역할을 한다. 이 개념은 지식, 특징 추출 및 열거(enumaration), 객체 상태의 구성 및 정의들을 일반화하는 기술을 제공하기도 한다. 그 자체를 구현하기 위해서는 일반적으로 그래프, 벡터 및 테이블 등의 공지된 데이터 구조를 이용할 수 있다.

또 다른 중요한 개념으로는 추론(reasoning)이 있다. 자동화된 추론은 지식 표현 및 인지를 이용하는 알고리즘을 결합하여 새로운 결론을 이끌어내고, 질문 및 답변을 추측해 내며, 에이전트 목표들(agent goals)을 달성하게 된다. 인공지능(AI) 추론의 예로는 다음과 같은 개념적 프레임워크들이 있다. 즉, 그 예로는 규칙 베이스(rule base)(추론을 위한 인지된 상태 및 지식베이스에 대해서 시스템 규칙들이 평가됨)와; 검색 시스템(이용 가능한 지식 및 목표에 따라 지식 결론의 검색을 위해 주지의 데이터구조를 이용함(예를 들어 결정 트리, 상태 그래프, 미니맥스 결정(minimax decision) 등이 그것에 포함됨))과; 분류기(classifier)(분류기 추론 시스템의 목적은 분류 태그(classification tag)가 없는 실험으로서 표현되는 인지 상태(perceived state)를 분류하기 위한 것임) 등이 있다. 미리 분류된(pre-classified) 지식베이스에 따라 분류기는 새로운 실험의 분류를 추측해낼 것이다(여기에는 예를 들어 벡터거리 발견적 방법(heuristics), 서포트 벡터 머신(Support Vector Machine), 분류기 신경회로망 등이 포함됨).

또 다른 중요한 개념으로 학습(learning)이 있다. 학습의 목표는 경험을 통한 일반화에 의해 인공지능 추론 시스템의 잠재적 성능을 향상시키는 것에 있다. 학습 알고리즘에 입력되는 것은 실험(experiment)이 될 것이고, 그 결과에 따라 지식베이스에 대한 수정들이 그 학습 알고리즘에서 출력될 것이다(여기에는 예를 들어 보강 학습, 배치학습, 서포트 벡터 머신 등이 포함됨).

인공지능 소프트웨어를 구현한 몇 가지 비한정적 예에는 다음과 같은 것들이 포함된다(아래 예들은 모두 "An Artificial Intelligence: A Modern Approach (인공지능: 최신 방법)", S. Russell and P. Norvig (eds), Prentice Hall, Pearson Education Inc., NJ, USA, 2003에 기재되어 있다):

**자율 플래닝 및 스케줄링(Autonomous planning and scheduling):** 미 항공우주국(NASA: 나사)의 원격 에이전트 프로그램은 우주선을 위한 동작들의 스케줄링을 제어하는 최초의 기내 탑재 자율 플래닝(on-board autonomous planning) 프로그램이 되었다. 원격 에이전트는 지상으로부터 특정된 고수준 목표들로부터 계획들을 생성하였고, 그 계획들이 수행됨에 따라 우주선의 동작을 감시하였다(문제들이 발생함에 따라 그 문제들을 검출, 진단 및 복구 함).

**게임 플레이(Game playing):** IBM의 디프 블루(Deep Blue)는 세계 챔피언을 이긴 최초의 컴퓨터 프로그램이 되었다.

**자율 제어:** ALVINN 컴퓨터 비전 시스템은 차선을 따라 차의 진행이 유지되게 운전하도록 훈련되어졌다. ALVINN은 이전의 훈련운행들(training runs)에서의 경험에 기초하여 운전하기에 가장 좋은 방향을 계산한다.

**진단:** 확률적 분석에 기초한 의료 진단 프로그램은 여러 의료분야에서 전문의 수준으로 수행이 가능하였다.

**물류 계획(logistics planning):** 1991년 걸프만 위기 동안에, 미군은 일명 DART라고 불리는 "동적 분석 및 재 계획 툴(Dynamic Analysis and Replanning Tool)"을 배치하여 운송을 위해 자동화된 물류 계획 및 스케줄링을 수행하였다.

**로봇공학:** 많은 외과 의사들이 미세 수술을 할 때 새로운 로봇 보조기를 이용한다. HipNav은 환자 내부에 대한 해부(anatomy)의 삼차원 모델을 생성하기 위해 컴퓨터 비전 기술을 이용하고, 히프 대체 보형물의 삽입을 안내하는데 로봇 제어를 이용하는 시스템이다.

**언어 이해 및 문제 해결:** RROVERB은 가능한 단어 채움 문자들에 있어서의 제약조건들과, 과거 퍼즐들의 많은 데이터베이스와, 여러 가지 정보소스들을 이용하여 사람보다도 쉽사말 퍼즐(crossword puzzle)을 더 잘 푸는 컴퓨터 프로그램이다.

소프트웨어용 유전적 알고리즘 및 진화 알고리즘을 위한 연구도 행해졌다. 이러한 소프트웨어의 하나의 예가 Karl Sims의 "Evolving Virtual Creatures"에 기술되어 있다(Computer Graphics, SIGGRAPH'94 Proceedings, 1994년 7월, 15-22쪽). 이 참고자료는 실제 물리적 세계의 모의 형태(simulated version)로서의 삼차원 가상세계에서 이동할 수 있는 "creature(생물)"이란 소프트웨어에 대해서 기술하고 있다. 이 생물들은 유전적 알고리즘을 이용하여 학습 및 진화할 수 있음으로써, 지시된 외부 입력(directed external input)이 없이도 그들의 행동을 변화시키게 된다. 따라서, 이들 유전적 알고리즘은 가상 세계에서의 서로 다른 "적합성(fitness)" 또는 보상들(rewards)을 갖는 잠재적 행동들의 초공간(hyperspace)을 묘사하였다. 이들 알고리즘 자체의 구현은 지시된 그래프들을 이용하여 이루어졌고, 이들에 의해 생물들의 유전자형들(구성성분들) 및 그들의 행동을 기술하였다.

그 모의(simulation)의 시작 시에, 다른 유전자형들을 갖는 많은 서로 다른 생물들이 모의되었다. 생물들은 가상세계에서 다른 자극에 대해서 그들의 행동을 변화시키도록 허용되었다. 각 "세대(generation)"에서는, 상대적 혹은 절대적 컷오프 점수(cut-off score)에 따라, 특정한 생물들만이 생존하도록 허용되며, 상기 점수는 생물들의 행동의 적합성(fitness)에 따라 결정되어진다. 돌연변이 발생이 허용되며, 이에 따라 돌연변이된 생물들의 적합성이 증가될 수도 있고, 혹은 그 반대일 수도 있다. 또한, 예를 들어, 노드와 관련된 값을 무작위로 변경하고, 그리고/또는 노드들을 추가 혹은 삭제함으로써 지시된 그래프를 통해 돌연변이가 수행되게 된다. 이와 유사하게, 생물들 사이의 "교배"에 따른 결과로, 지시된 그래프(directed graph)에 변화들이 생길 수 있다.

상술한 참고자료에 기술되어 있는 결과들은 가상생물들이 사실상 변화 및 진화할 수 있다는 것을 보여주고 있다. 하지만, 그 생물들은 가상 세계 내에서만 동작할 수 있고, 실제 물리적 세계와, 및/또는 인간 컴퓨터 조작자들과는 어떠한 연결점 또는 관련점도 가지고 있지 않다.

### 발명이 이루고자 하는 기술적 과제

이러한 배경기술은 특히 인간 사용자들과 상호작용을 하기 위해 이동 정보 장치용의 지능형 소프트웨어가 학습 및 진화할 수 있도록 하는 시스템 또는 방법을 적어도 제시하거나 제안하지 못하고 있다. 또한, 배경기술은 사전행동적(proactive) 사용자 인터페이스가 사용자의 행동을 학습하여 사용자에게 선택사항들을 능동적으로 제안 할 수 있는 연산장치용 사전행동적 사용자 인터페이스에 대해서도 제시하거나 제안하지 않고 있다. 또한, 배경기술은 사용자 인터페이스가 사용자의 행동에 따라 능동적으로 변화하게 되는 이동 정보 장치용 적응형 시스템에 대해서도 제시하거나 제안하지 않고 있다. 또한, 배경기술은 아바타를 통해 인간 사용자와 상호작용을 할 수 있는 이동 정보 장치용 지능형 에이전트에 대해서도 제시하거나 제안하지 않고 있다.

### 발명의 구성 및 작용

본 발명은 어떠한 형태의 연산장치에도 임의로(optionally) 설치될 수 있는 (또는 그렇지 않을 경우 그 장치를 제어 할 수 있는 및/또는 그 장치와 연관되어질 수 있는) 사전행동적 사용자 인터페이스를 제공함으로써 상술한 배경기술의 문제점을 해결하고자 하는 것이다. 이러한 사전행동적 사용자 인터페이스는 특정 사용자와의 이전 경험에 기초하여, 및/또는 사용자의 행동에 따라 연산장치가 선택할 수 있는 여러 가지 미리 프로그램된(pre-programmed) 패턴들에 기초하여, 사용자에게 능동적으로 제안들을 하게 되며, 및/또는 그렇지 않다면 결정되지 않은 또는 예상되지 않은 행동(non-deterministic or unexpected behavior)에 관여하게 된다. 예를 들어, 그 일부 또는 메뉴를 바꾸고; 표시하기 위한 다른 메뉴들을 제공하고; 및/또는 터치스크린 기능을 바꾸는 등, 표시부의 적어도 일부의 외관을 바꿈으로써 상술한 제안들이 임의로 행해질 수 있다. 또한, 그러한 제안들은 음성에 의해 임의로 행해질 수도 있다. 그 밖의 다른 형태의 제안들 또는 전달 메커니즘(delivery mechanism)들도 가능하다.

"제안"에 의해, 상술한 시스템은 주어진 소정의 사용자 성향(preferences)을 고려하여 또한 시스템의 상태가 행위의 특정한 수행을 허용하는지 여부에 따라 그 행위를 사실상 임의로(optional) 수행할 수 있음을 유념해야 한다.

일반적으로, 사전행동적 사용자 인터페이스는 적어도 지능적이고 쌍방향(interactive)으로 보이는 것이 바람직하고, 사용자와 적어도 어느 정도 "자유로운"(예를 들어, 준비되지 않은(non-scripted) 혹은 부분적으로 준비된) 통신을 할 수 있는 것이 바람직하다는 점을 강조하는 것은 중요하다. 사용자의 기대들(expectations)이 "지능형" 에이전트/장치와의 상호작용을 위해 이행되는 것이 바람직하다는 의미에서, 지능적인 모습(intelligent appearance)을 갖는 것은 중요하다. 이러한 기대들은 통신할 수 있는 능력, 인터페이스의 선택적 외관(optional appearance), 의인화 속성(들)(anthropomorphic attribute(s))의 사용 등의 요소들에 의해 임의로 정형화(shaped)될 수 있으며, 그것들은 사전행동적 사용자 인터페이스와 사용자 사이의 상호작용에 있어서 지능 감각(sense of intelligence)을 향상시키는데 이용되는 것이 바람직하다. 사용자로부터 전달되는 통신에 있어서, 사전행동적 사용자 인터페이스는 사용자가 이동 정보 장치와 어떠한 방식으로 상호작용하기를 원하는지를 감지할 수 있는 것이 바람직하다. 임의로, 통신(정보전달)은 한쪽 방향으로만 이루어질 수 있다. 예를 들어, 인터페이스는 임의로 사용자에게 메시지 혹은 정보를 제시하지만, 사용자로부터는 정보를 수신하지 않을 수 있으며, 혹은 그 반대로 구현하는 것도 가능하다. 통신은 사용자와의 바람직한 상호작용을 위해 쌍방향으로 이루어지는 것이 바람직하다.

임의로 및 바람직하게, 사용자로 전달되는 통신을 위해, 사전행동적 인터페이스는 사용자와의 통신의 일부로서 사용자와의 상호작용을 위해 모의된 감정들(simulated emotions)을 표시하거나 보여줄 수 있는 기능을 갖는다. 이후 더욱 자세히 설명되는 바와 같이, 이를 감정들은 지능형 에이전트에 의해 제시하기 위해 모의되는 것이 바람직하며, 아바타 혹은 생물(creature)에 의해 표현되어지는 것이 더욱 바람직하다. 그 감정들은 감정형 시스템(emotional system)을 통해 생성되는 것이 바람직하며, 임의로 적어도 하나의 사용자 성향(user preference)에 따라 부분적으로 제어될 수 있다. 지능형 시스템은 지능형 에이전트의 반응들 및 통신들이 사용자가 인지하는데 있어서 믿을 만한 것이 될 수 있도록 하는데 이용하는 것이 바람직하다. 예를 들어, 지능형 에이전트가 개와 같은 생물로서 제시되면, 감정형 시스템은 그 감정들이 "개와 같은" 행동에 대한 사용자의 기대와 일치될 수 있도록 하는 것이 바람직하다.

이와 유사하게, 지능형 에이전트는 사용자에 대해서 적어도 지능적으로 보이는 것이 바람직하다. 임의로, 그 지능(intelligence)은 완전히 결정적인 메커니즘(completely deterministic mechanism)을 통해서 제공될 수 있지만, 지능적인 모습을 갖추기 위한 기초가 되는 것으로서 적어도 하나 이상의 무작위 혹은 반-무작위 요소들(semi-random elements)이 포함되는 것이 바람직하다. 또한, 그러한 요소들은 지능형 에이전트의 표현(representation)에 대해서 지능과 관련한 사용자의 기대와 일치할 수 있도록 제시되는 것이 바람직하다.

사용자의 요구 혹은 다른 통신을 만족시키기 위해 적어도 어느 정도는 지능형 에이전트가 행동을 변경하는 것이 가능하도록 하기 위해서 적응특성 혹은 적응기능(adaptiveness)이 제시되는 것이 바람직하다. 사전행동적 사용자 인터페이스는 사용자와 통신하기 위한 지능형 에이전트를 포함하고 있지 않더라도, 적응특성에 의해 인터페이스가 사전행동적이 될 수 있도록 하는 것이 바람직하다. 사용자의 이동 정보 장치와의 상호작용의 관찰(observation)에 의해 그러한 적응기능이 실행될 수 있도록 하는 것이 바람직하며, 그렇지만 임의로 및 바람직하게 그러한 관찰에 대한 사전행동적 사용자 인터페이스의 반응이 지식베이스 및/또는 규칙베이스에 의해 안내될 수도 있다.

그러한 적응기능의 비한정적이지만 바람직한 특정 예로서, 특히 복수의 메뉴를 포함하는 이동 정보 장치에 있어서, 그러한 적응기능에는 메뉴의 적어도 하나의 형태(aspect)를 변경할 수 있는 능력이 포함될 수 있는 것이 바람직하다. 예를 들어, 임의로 하나 이상의 단축키 혹은 단축수단(shortcuts)이 제공되어, 사용자가 메뉴 계층(hierarchy)에 있어서 마지막 선택대상(final choice)보다 상위에 있는 서브메뉴 혹은 이전 메뉴 중 적어도 하나(더욱 바람직하게는 그 모두)를 우회하여 직접 메뉴 선택대상에 도달할 수 있도록 할 수 있다. 임의로 (그렇게 하는 대신에 혹은 부가적으로), 하나 이상의 메뉴가 적응형 사용자 인터페이스의 적응기능에 따라 예를 들어 사용 빈도수에 따라 재배열될 수 있다. 임의로, 그러한 재배열하는 것에는 메뉴 선택 및/또는 서브메뉴와 같은 메뉴의 일부를 메뉴 계층에 있어서 현재 위치보다 상위에 있는 새로운 위치로 이동시키는 것이 포함될 수 있다. 메뉴 계층에 있어서 상위에 있는 서브 메뉴는 하위 위치(보다 아래쪽에 있는 위치)에 있는 것들보다 적은 횟수의 메뉴 선택을 통해 보다 빨리 도달될 수 있다.

임의로 및 바람직하게, 적응기능 및/또는 감정들(emotions)은 사전행동적 사용자 인터페이스에 의해 학습을 위한 보상들(rewards)을 통해 도움을 받을 수 있다. 사용자가 승인한 제안들 혹은 행위들은 보상 혹은 긍정적 인센티브를 사전행동적 인터페이스에게 제공하여 그러한 제안들 혹은 행위들(actions)을 계속하도록 하는 것이 바람직하며, 사용자에 의해 승인되지 않으면 사전 행동형 인터페이스에게 그러한 행동(들)(behavior(s))을 계속하는 것에 대한 행동 억제 요인(disincentive)을 발생시키는 것이 바람직하다. 사전행동적 사용자 인터페이스에게 긍정적 혹은 부정적 인센티브들/행동억제요인들을 제공함으로써, 행동이 허용 혹은 금지되는 보다 "흑백(black or white)"적인 방식에 비해, 그 인터페이스의 행동에 뉘앙스가 더욱 부여되도록 하는 것이 바람직하다. 그러한 행동들이 적어도 어느 정도 사용자에 의해 일괄적으로 승인/부인될 경우, 그러한 뉘앙스에 의해 반대 혹은 모순되는 행동들(opposing or contradictory behaviors)이 다루어질 수 있도록 하는 것도 바람직하다.

상기 사전행동적 사용자 인터페이스의 또 다른 임의의 그러나 바람직한 기능에는 사용자에 대한 교육기능이 포함된다. 사용자에게 이동 사용자 장치의 기능들에 대해서 알려주기 위해 그러한 교육이 임의로 수행될 수 있다. 예를 들어, 사용자가 예를 들어 잘못된 선택을 입력하여 그 장치를 올바로 조작하지 못하면, 상기 교육 기능은 사용자가 그 장치를 올바로 사용하는 방법을 학습할 수 있도록 도와주는 것이 바람직하다. 하지만, 상기 교육 기능이 적어도 하나의 비장치 관련 대상(non-device related subject)에 대해서 사용자에게 명령(instruction)을 제공할 수 있는 것이 더욱 바람직하다. 상기 교육 기능의 바람직한 실시 예에 따르면, 임의로 및 바람직하게 (더욱 바람직하게는 유연성 애플리케이션 프레임워크(flexible application framework)를 통하여) 복수의 대상들에 대해서 (혹은 비장치 관련 대상을 변경하는 것에 의해) 명령이 제공될 수 있다.

본 발명의 임의의(optional) 그러나 바람직한 실시 예에 따르면, 사전행동적 사용자 인터페이스가 사용자와 상호작용을 하여 사용자에 대한 모델이 구성되는 것이 바람직하다. 그러한 모델은 프로그램된 및/또는 사용자의 행동으로부터 결정된 인공지능 지식베이스들을 임의로 및 바람직하게 통합하게 될 것이다. 또한, 상기 모델은 사전행동적 사

용자 인터페이스가 사용자 인터페이스에 의해 행해진 특정한 제안들에 대한 사용자의 반응을 측정하는 것을 임의로 가능하게 함으로써, 사용자의 내재적 혹은 비명시적 성향(implicit preferences)에 적응하게 된다.

그러한 연산장치들의 비한정적 예에는, ATM(예를 들어 사용자 행동의 특정 패턴이 경보기를 작동하게 할 수 있으므로, 보안성과도 관련됨)와, 모든 타입의 일반 컴퓨터(데스크탑, 랩탑, 소형 경량 및 착용가능 컴퓨터 등)와, 휴대폰 등의 이동 정보 장치, 페이저장치, 그 이외의 무선 통신장치, 운영체제를 갖춘 일반 전화기, PDA 및 무선 PDA, 운영체제를 갖춘 소비재 등이 포함된다. 이하, "연산장치"라는 용어는 운영체제를 갖추고 있고 연산을 수행할 수 있는 모든 전자 장치를 포함하는 것으로 한다. 운영체제는 임의로 내장된 시스템 및/또는 다른 타입의 소프트웨어 및/또는 하드웨어 런타임 환경(runtime environment)일 수 있다. 이하, "이동 정보 장치"라는 용어는 모든 타입의 무선 통신장치(여기에는 휴대폰, 무선 페저, 무선 PDA 등이 포함되며 이것들에 한정되는 것은 아니다)를 포함하는 것으로 간주되지만, 그것에 한정되는 것은 아니다.

바람직하게는, 본 발명은 현재의 포괄적(generic)이고 비유연적(non-flexible)인 사용자 인터페이스를 유연성이 있고 실제적으로 사용자 친화적인 인터페이스로 변화시킬 뿐만 아니라, 연산장치와의 향상된 사용자 경험 및 상호작용을 제공하기 위해 구현된다. 좀 더 바람직하게는, 본 발명은 사용자와 상호작용하게 되는 아바타 형태로 사용자인 인터페이스를 구성하는 임의의 그러나 바람직한 실시예에 따라, 연산장치와의 사용자의 향상된 감정적 경험(emotional experience)을 제공하기 위해 구현된다. 따라서, 본 발명은 특히, 예를 들어 휴대폰과 같은 이동 정보 장치에 "생존형 장치(living device)"의 경험을 부여할 수 있다. 이 실시예에 따르면, 사용자는 "생존형 장치"에 감정적 부착기능(emotional attachment)을 형성할 수도 있다.

본 발명의 다른 실시예에 따르면, 적응형 시스템을 포함하는 이동 정보 장치가 제공된다. 상술한 사용자 인터페이스처럼, 이 장치도 프로그램된 패턴 및/또는 사용자와의 이전 경험에 의존하게 된다. 하지만, 임의로 및 바람직하게, 적응형 시스템은 이동 정보 장치의 기능들 및 환경 내에서의 동작에 더욱 제한되어 있다.

임의로 및 바람직하게, 이동 정보 장치 적응형 시스템 및 사전행동적 사용자 인터페이스 중 어느 하나 혹은 그 양쪽 모두는 유전적 알고리즘, 인공지능(AI) 알고리즘, 기계학습(ML: Machine Learning) 알고리즘, 학습된 행동, 및 진화할 수 있는 소프트웨어/연산장치로 구현되게 된다. 또한, 상술한 두 가지 중 어느 하나 혹은 그 모두는 고급수준의 음성 명령들, 터치스크린 명령들, 키보드 '단축키 혹은 단축수단(short-cuts)'들을 임의로 제공할 수도 있다.

본 발명의 또 다른 임의의 그러나 바람직한 실시예에 따르면, 이동 정보 장치 망(network)을 거쳐 이동 정보 장치와 사용하기 위한 하나 이상의 지능형 에이전트가 제공되며, 이것에는 에이전트가 인간 사용자와 통신(정보교환)을 할 수 있게 하는 아바타(혹은 "생물"; 이하 이 두 용어를 서로 치환 가능한 용어로서 사용하기로 함)가 포함되는 것이 바람직하다. 따라서, 아바타는 사용자와의 상호작용을 위한 사용자 인터페이스를 제공하는 것이 바람직하다. 또한, 지능형 에이전트는 네트워크를 통해 이동 정보 장치의 적어도 하나의 상호작용을 제어하기 위한 에이전트를 포함하는 것이 바람직하다. 이 실시예는 임의로 이동 정보 장치 망을 거쳐 연결되는 그러한 지능형 장치를 복수 개 포함하여, 임의로 그러한 에이전트들의 네트워크를 형성할 수도 있다. 이러한 실시예를 통해 여러 가지 응용예들이 임의로 제공될 수도 있으며, 그 응용예에 포함되는 것들은 예를 들어 일반적인 교육(teaching) 및/또는 특히 이동 정보 장치를 사용하는 방법을 학습하기 위한 교육, 언어 교육, 통신 애플리케이션, 커뮤니티 애플리케이션, 게임, 오락(entertainment), 쇼핑(쿠폰 획득 등), 가게 또는 다른 장소 위치파악, 광고 및 원하지 않는 메시지의 필터링, 롤플레잉 게임 또는 휴대전화 망을 통한 그 이외의 쌍방향(interactive) 게임, "채팅" 및 미팅 기능, 지능형 에이전트를 위한 "선물(presents)"을 사거나 그렇지 않으면 그 캐릭터를 장식하기 위한 기능 등이 있으며, 이것들에 한정되는 것은 아니다. 이론적으로는, 이러한 에이전트들 자체에 액세서리로서 "애완동물(pet)"이 제공될 수도 있다.

그러한 지능형 에이전트는 임의로 휴대폰 조작자들에게 다양한 사업/촉진용 기회들을 제공하는데 도움을 줄 수도 있다. 임의로 및 바람직하게, 상기 에이전트들은 휴대폰 상의 소프트웨어를 설치 및 구동시키는데 도움을 줄 수도 있으며, 이는 새로운 산업분야에 해당한다. 예를 들어, 임의로 에이전트는 소프트웨어를 올바로 다운로드하고 구동시키기 위해 필수적인 이동 정보 장치의 적절한 형태와 그 이외의 구체 사항들을 결정하는데 도움을 줄 수도 있다.

임의로 및 바람직하게 지능형 에이전트는 사용자에게 이동 정보 장치 자체에 있는 다양한 기능들을 동작시키는 방법, 예를 들어 메시지의 송신 또는 수신, 또는 알람을 이용하는 등의 방법을 가르침으로써 사용자를 교육시킬 수도 있다. 이후 더욱 상세히 설명하는 바와 같이, 그러한 교육 기능은 임의로 이동 정보 장치 외부의 정보/기능들에 대해서 사용자에게 가르치도록 확장될 수도 있다. 바람직하게, 그러한 교육 기능들은 네트워크 내에서의 복수의 에이전트들 사이의 통신에 의해 향상됨으로써 네트워크 상의 에이전트 사이에 분포되어 있는 정보를 에이전트가 획득할 수 있게 된다.

임의로 및 바람직하게, 에이전트를 위한 결제(지불)는 정기 사용료 지불(subscriptions) 방식으로 행해질 수 있으며, 혹은 그 대신에 사용자의 선불잔금에 부과되고 및/또는 월말에 사용자에게 청구되는 행위들(actions)을 통해 임의로 에이전트에게 "지급(feed)"될 수 있다.

따라서, 본 발명의 다양한 실시예에 따르면 많은 서로 다른 상호작용이 가능하다. 이러한 상호작용들 중에는, 아바타 혹은 그 이외의 캐릭터 또는 사용자의 의인형 상징체(personification)와 장치 사용자와 사이의 상호작용; 아바타 또는 그 이외의 캐릭터 또는 의인형 상징체를 통해 장치를 구동하기 위한, 장치와 장치사용자와 사이의 상호작용; 아바타 또는 그 이외의 캐릭터 또는 의인형 상징체를 통한 통신(정보교환)에 의해, 각각의 장치를 이용한 두 사용자 사이의 상호작용; 및 임의로 사용자들 사이의 또는 심지어는 에이전트와 사용자 사이의 어떠한 정보교환도 없이, 두 장치 간에 각각의 지능형 에이전트를 통한 상호작용 중 어느 하나 이상이 포함된다. 상기한 상호작용 또는 가능한 상호작용들은 본 발명의 실시예에 따라 결정되며, 이것에 대해서는 아래에 더욱 상세히 설명한다.

본 발명은 연산장치 및/또는 예를 들어 휴대폰과 같은 이동 정보 장치의 상대적으로 제한된 환경으로부터 이의를 얻게 되는데, 그 이유는 그러한 환경의 변수들이 이미 알려져 있기 때문이다. 그러한 장치들은 예를 들어 휴대전화망과 같은 네트워크를 통해 통신을 하더라도, 여전히 상기 환경의 변수들은 미리 결정될 수 있다. 현재, 연산장치는 사

용자의 수동적 및 직접적인 조작으로도 고객 맞춤화가 허용되지 않거나 그것이 거의 허용되지 않는 포괄적 인터페이스(generic interface)만을 제공하고 있다.

"소프트웨어"라는 용어는 하드웨어에 의해 작동하는 프레임워크 또는 명령어들을 임의로 포함하는 것일 수도 있다 는 점을 유념해야 한다.

#### (바람직한 실시예)

본 발명은 사전행동적 사용자 인터페이스에 관한 것으로 어떠한 형태의 연산장치에도 임의로 설치될 수 있다(혹은 그렇지 않을 경우에 그 장치를 제어할 수 있거나 및/또는 그 장치와 관련될 수 있다). 사전행동적 사용자 인터페이스는 사용자 행동에 따라, 연산장치가 선택할 수 있는 다양한 미리 프로그램된 패턴들 및/또는 특정한 사용자와의 이전 경험에 기초하여, 사용자에게 제안들을 능동적으로 행할 수 있다. 표시부의 적어도 일부의 외관을 바꿈으로써, 예를 들어, 그 일부 또는 메뉴를 바꾸고; 표시하기 위한 다른 메뉴들을 제공하고; 및/또는 터치스크린 기능을 바꾸는 등에 의해 상술한 제안들이 임의로 행해질 수 있다. 또한, 그러한 제안들은 음성에 의해 임의로 행해질 수도 있다.

사전행동적 사용자 인터페이스는 상술한 바와 같이 운영체제를 포함하고 있는 연산장치를 위해 구현되는 것이 바람직하다. 이러한 인터페이스는 임의로 및 바람직하게 사용자와 운영체제 사이의 통신을 위한 사용자 인터페이스를 포함한다. 또한, 바람직하게 인터페이스는 사용자 인터페이스와의 사용자의 상호작용의 적어도 하나의 패턴을 검출하고 검출된 패턴에 따라 사용자 인터페이스의 적어도 하나의 기능을 사전행동적으로 변경하기 위한 학습모듈을 포함한다. 이에 따라, 사전행동적 사용자 인터페이스는 사용자의 요청을 예측하여 연산장치의 원하는 기능을 선택할 때 사용자에게 도움을 줄 수 있다.

임의로 및 바람직하게, 사용자 인터페이스와 사용자의 이전의 상호작용들 중 적어도 하나에 따라 결정된 패턴과, 미리 결정된 패턴 또는 그 조합으로 이루어진 군으로부터 적어도 하나의 패턴이 선택되어진다. 제1 패턴 타입은 학습된 행동을 표현하며, 제2 패턴 타입은 임의로 프로그램되어질 수 있거나, 그렇지 않으면 미리 결정될 수 있으며, 이는 특히 특정 연산장치가 사용자에 의해 처음으로 사용될 경우 사용자를 돋기 위한 것이다. 제3 패턴 타입은 이러한 두 가지 요소를 조합하게 되며, 패턴이 사용자 행동에 따라 적어도 부분적으로(완전하게는 아님) 결정될 수 있도록 하며, 예를 들어, 패턴 선택은 복수의 규칙들에 따라, 및/또는 가능한 세상 환경상태 및/또는 장치 및/또는 사용자 인터페이스의 상태의 제한적 정의에 따라 임의로 안내될 수 있다(더 구체적인 것은 아래의 설명을 참고).

사용자 인터페이스는 그래픽 표시를 특정적 요소로서 가지고 있어, 그 그래픽 표시 중 적어도 하나의 기능이 패턴에 따라 사전행동적으로 변경되도록 하는 것이 바람직하다. 예를 들어, 임의로 및 바람직하게, 그래픽 표시 중 적어도 일부가 변경될 수 있으며, 더욱 바람직하게는 검출된 패턴에 따라 표시할 메뉴를 선택하여 표시하는 것에 의해 그러한 변경이 행해질 수 있다. 임의로, 그러한 메뉴는, 예를 들어 메뉴를 "수행중(on the fly)"에 생성하기 위해, 복수의 메뉴 옵션들로부터 하나의 메뉴를 구성함으로써 선택될 수 있다.

사용자 인터페이스는 추가적으로 혹은 그 대신에 음성표시를 특징으로서 가져서, 사용자 인터페이스의 적어도 하나의 기능을 변경하는 것이 연산장치에 의해 생성된 적어도 하나의 가정음을 변경하는 것을 포함하도록 할 수도 있다.

임의로 및 바람직하게, 사전행동적 사용자 인터페이스는 본 발명의 방법에 따라 구현될 수 있고, 이는 바람직하게 사용자와 연산장치 사이의 사용자 인터페이스를 통한 사전행동적 상호작용을 위해 구현되어진다. 상술한 방법은 사용자 인터페이스와 사용자의 적어도 하나의 상호작용에 따라 사용자 행동의 패턴을 검출하는 것과, 그 패턴에 따라 사용자 인터페이스의 적어도 하나의 기능을 사전행동적으로 변경하는 것을 포함한다.

본 발명의 다른 실시예에 따르면, 적응형 시스템을 포함하는 이동 정보 장치가 제공된다. 상술한 사용자 인터페이스와 같이, 이 장치도 사용자와의 이전 경험 및/또는 미리 프로그램된 패턴들에 의존하게 된다. 하지만, 임의로 및 바람직하게, 그러한 적응형 시스템은 현재 PDA로부터의 특정 기본 기능들을 포함하기도하는 예를 들어 휴대폰과 같은 이동 정보 장치의 환경 및 기능들 내에서의 구동에 더욱 한정된다.

바람직하게, 상기 적응형 시스템은 운영체제를 특징적 요소로서 갖춘 이동 정보 장치와 함께 동작한다. 이 운영체제는 임의로 내장형 시스템(embedded system)을 포함할 수 있다. 이동 정보 장치는 휴대폰을 임의로 포함할 수 있다.

바람직하게, 적응형 시스템은 이동 정보 장치와 사용자의 복수의 상호작용을 분석함으로써 사용자 행동을 분석할 수 있게 되며, 더욱 바람직하게, 그 후에 적응형 시스템은 적어도 하나의 이미 결정된 패턴과 복수의 사용자의 상호작용을 비교하여 그 미리 결정된 패턴이 사용자 인터페이스의 적어도 하나의 기능을 변경하는 것에 연관되는지 여부를 알게 된다. 그 대신에 혹은 추가적으로, 상기한 분석은 이전에 검출된 사용자 행동의 적어도 하나의 패턴과 상술한 복수의 사용자의 상호작용을 비교하는 것을 포함할 수 있으며, 이때 상술한 이전에 검출된 사용자 행동의 패턴은 사용자 인터페이스의 적어도 하나의 기능을 변경하는 것과 관련된다.

임의로, 사용자 인터페이스의 기능은 이동 정보 장치에 의해 가정음을 생성하는 것을 포함할 수 있고, 이것은 좀더 바람직하게 벨소리(ring tone)와, 알람소리와, 수신 메시지 소리 중 적어도 하나로 이루어진 군으로부터 선택되어진다. 그 대신에 혹은 추가적으로, 이것은 이동 정보 장치에 의한 화상표시와 임의로 관련될 수 있다. 이 화상표시는 예를 들어 메뉴를 표시하는 것을 임의로 포함할 수 있다.

반드시 그런 것은 아니지만, 적응형 시스템은 이동 정보 장치 자체에 의해 임의로 구동될 수 있다. 혹은, 이동 정보 장치가 네트워크에 연결된 경우, 적응형 시스템은 적어도 부분적으로 네트워크로부터 이동 정보 장치로 송신된 명령들에 따라 임의로 구동될 수도 있다. 이것을 구현하기 위해서는, 적응형 시스템의 적어도 하나의 동작과 관련된

데이터가 이동 정보 장치와 다른 위치에 저장되는 것이 바람직하며, 이때 상기 위치는 네트워크를 통해 접근이 가능하다.

본 발명의 바람직한 실시예에 따르면, 적응형 시스템은 수신된 입력정보 및 이전에 얻어진 지식에 따라 분석을 행하는 학습 모듈을 또한 포함한다. 임의로(optional), 이러한 지식은 사용자의 행동으로부터 이전에 얻어진 것일 수 있고, 및/또는 특정 이동 정보 장치의 적응형 시스템과 통신을 하는 다른 적응형 시스템으로부터 전달된 것일 수 있다. 임의로 및 바람직하게, 적응형 시스템은 인공지능 알고리즘과, 기계 학습 알고리즘, 또는 유전적 알고리즘 중 하나 이상에 따라 사용자 행위에 적응된다.

본 발명의 또 다른 임의의 그러나 바람직한 실시예에 따르면, 이동 정보 장치 망을 통해 이동 정보 장치와 사용하기 위한 하나 이상의 지능형 에이전트가 제공되며, 이것에는 에이전트가 인간 사용자와 정보교환(통신)을 가능하게 하는 아바타가 포함되는 것이 바람직하다. 따라서, 바람직하게 아바타는 사용자와의 상호작용을 위한 사용자 인터페이스를 제공한다. 또한 지능형 에이전트는 네트워크를 통하여 이동 정보 장치의 적어도 하나의 상호작용을 제어하기 위한 에이전트를 포함하는 것이 바람직하다. 이 실시예는 이동 정보 장치 망을 통해 연결된 그러한 복수의 아바타를 임의로 포함할 수 있다.

본 발명의 바람직한 실시예에 의하면, 아바타 및 에이전트는 모두 이동 정보 장치에 의해 구동되어진다. 그 대신에 혹은 추가적으로, 이동 정보 장치는 제2의 에이전트를 갖는 적어도 하나의 다른 이동 정보 장치와 통신을 하여, 그 제1 및 제2 에이전트가 서로 통신이 가능하게 되는 것이 바람직하다. 임의로, 이러한 통신은 직접 행해질 수 있는데, 예를 들어 2개의 이동 정보 장치 사이에서 적외선 통신을 통해 직접 행해질 수 있고, 혹은 그 대신에 또는 추가적으로 이동 정보 장치 망을 통해 행해질 수 있다. 예를 들어, 그 네트워크가 휴대 전화망일 경우, 통신은 표준 통신 프로토콜, IP/HTTP, SMS 등을 이용하여 임의로 수행될 수 있다.

또한, 임의로 및 더욱 바람직하게, 각 이동 정보 장치의 사용자들은 그들 각각의 아바타를 통해 통신할 수 있는 것이 바람직하다. 이러한 통신은 룰플레잉 게임과 같은 게임과 임의로 관련될 수 있다.

상술한 적응형 시스템과 유사하게, 아바타 및 에이전트 중 하나 또는 그 모두는 적어도 부분적으로 이동 정보 장치 망으로부터 이동 정보 장치로 송신된 명령들에 따라, 임의로 동작될 수 있다. 또한, 임의로, 아바타 또는 에이전트 중 적어도 하나의, 하나 이상의 동작과 연관된 데이터는 이동 정보 장치 이외의 위치에 저장되며, 상기 위치는 이동 정보 장치 망을 통해 접근이 가능하다.

본 발명의 바람직한 실시예에 따르면, 아바타의 모습의 적어도 하나의 특성이 예를 들어 사용자 명령에 따라 임의로 변경이 가능한 것이 바람직하다. 임의로 및 더욱 바람직하게, 아바타의 모습의 복수의 특징들이 미리 정의된 아바타 피부에 따라 변경 가능하다. 그 피부는 사용자에 의해 임의로 미리 결정되어진다. "피부"에 의해, 상기한 복수의 특징들이 하나의 세트로서 함께 변경된다는 것을 의미하며, 이 때 상기 세트는 상기 피부를 형성하는 것이다. 이 실시예가 네트워크-접근 가능 위치에 저장된 아바타와 관련된 데이터의 적어도 일부를 갖는 이전의 실시예와 결합될 경우에는, 사용자는 동일한 아바타를 서로 다른 전화기들로 임의로 이동시킬 수 있고, 및/또는 서로 다른 이유들 때문에, 예를 들어 파티 혹은 그 이외의 축제(celebration)와 같은 특별한 경우들을 위해, 아바타의 모습을 고객 맞춤화 할 수 있다. 물론 이러한 것들은 단지 예를 들기 위한 것으로 어떠한 방식으로도 한정적인 것을 의미하지는 않는다.

본 발명의 또 다른 임의의 그러나 바람직한 실시예에 따르면, 아바타의 모습의 적어도 하나의 특성은 자동화된 진화 알고리즘, 예를 들어 유전적 알고리즘에 따라 변경이 가능한 것이 바람직하다. 진화 알고리즘은 사용자를 위해 아바타의 개별 맞춤화(personalization)를 제공하기 위한 방법의 하나의 비한정적 예이다. 개별 맞춤화는 하나 이상의 특성들 또는 피부들(특성 그룹들)에 대한 직접적인 사용자의 선택을 통해 임의로 행해질 수도 있다. 그러한 개별 맞춤화는 적어도 부분적으로는 아바타로, 그에 따라 이동 정보 장치로, 사용자의 감정적 표현을 향상시키기 때문에 바람직한 것이다.

본 발명의 또 다른 임의의 그러나 바람직한 실시예에 따르면, 이동 정보 장치 망은 이동 정보 장치의 물리적 위치를 결정하기 위한 위치결정기(locator)를 포함하여, 바람직하게 사용자가 에이전트의 행위를 통해 그 물리적 위치에 대한 정보를 요구할 수 있다. 또한, 바람직하게, 위치결정기는 이동 정보 장치의 물리적 위치에 대해 상대적인 제2 물리적 위치를 결정할 수 있어, 사용자가 에이전트의 행위를 통해 제2 물리적 위치에 대한 정보를 요구할 수 있다.

임의로 및 바람직하게, 사용자는 상업적 위치, 의료적 위치, 및 예를 들어 소방서 또는 경찰서 등의 공공 안전 위치로 이루어진 군으로부터 임의로 선택될 수 있는 카테고리에 따라 제2 물리적 위치를 선택 및 요청 할 수 있다.

예를 들어, 사용자가 상업적 위치를 찾고자 할 때에는, 사용자는 가게 및 쇼핑몰 등의 이름에 의해 임의로 그려한 위치에 대한 요청을 입력하며, 혹은 그 대신에 가게 타입 등의 상업적 위치의 타입에 따라, 혹은 예를 들어 "쇼핑몰" 등의 일반 카테고리에 따라서도, 그러한 위치에 대한 요청을 입력하게 된다. 임의로, 일치하는 상업적 위치는 상기에 이전트의 상기 행위에 따라 이동 정보 장치로 메시지를 전송될 수 있고, 임의로 및 바람직하게, 이것에는 광고 또는 쿠폰 또는 그 조합 중 적어도 하나가 포함된다. 바람직하게, 에이전트는 그러한 메시지들에 대해, 혹은 그 모든 타입의 메시지에 대해, 적어도 하나의 판단기준에 따라 메시지를 필터링하며, 더욱 바람직하게 그 판단기준은 아바타를 통해 사용자에 의해 입력되며, 및/또는 메시지 수신시에 사용자가 이전에 한 행위에 응답하여 아바타에 의해 학습되어진다. 그리고, 아바타는 (전체적으로 혹은 부분적으로) 메시지 자체를 보여주지 않을 경우에는, 적어도 메시지에 대한 정보를 사용자에게 임의로 보여줄 수 있다.

또한, 바람직하게, 사용자는 아바타를 통해 제2 물리적 위치에 대한 정보를 요구한다.

물론, 상업적 위치는 반드시 물리적 위치일 필요는 없고, 예를 들어, 임의로, 이동 상거래(m-commerce) 등을 위한 가상의 상업적 위치일 수도 있으며, 이 경우, 사용자는 아바타를 통해 가상의 상업적 위치와 통신하게 된다. 바람직하게, 사용자는 아바타를 통해 가상의 상업적 위치에서 구입을 행할 수 있다. 또한, 사용자는 아바타를 다시 인터페

이스로서 이용하는 것이 바람직하지만, 에이전트를 이용함으로써 가상의 상업적 위치를 임의로 탐색할 수도 있다. 상술한 바와 같이 아바타는 가상의 상업적 위치로부터 구입한 액세서리를 임의로 전달받을 수도 있다.

또한, 이동 정보 장치가 소프트웨어를 수신할 수 있으면, 에이전트는 이동 정보 장치 상의 소프트웨어의 설치 중 적어도 일부를 행하는 것이 바람직하다. 사용자는 소프트웨어의 환경설정(configuration) 중 적어도 일부를 수행하기 위해 아바타와 임의로 상호작용할 수도 있다.

기술적 구현(technical implementation)에 있어서는, 본 발명은 사용자에게 매우 퍼스널한 장치의 (메모리, 데이터 처리 용량, 스크린 표시 크기 및 해상도 등에 있어서의) 제한된 시스템 상에서 동작할 수 있는 것이 바람직하다. 예를 들어, 바람직하게 상기 장치는 휴대폰 등의 이동 정보 장치이며, 이것은 꽤 휴대성 및 사용상 편리성을 위해 적응되므로, 상기 장치는 상술한 제한들 중 하나 이상 혹은 그 모두를 가질 수 있다. 본 발명의 구현 형태들은 이러한 특성들의 조합에 적합하도록 하는 것이 바람직하다. 따라서 사용자의 바람직한 개별맞춤화(personalization) 및 "개인적인 느낌(personal feel)"을 유지하면서도 상기 장치 자체의 제한들을 해결하기 위해, 다음과 같은 여러 가지 해결방안들을 제안하도록 한다. 이들 해결방안들은 단지 예에 불과하며 어떠한 식으로도 한정적으로 해석하도록 의도된 것은 아님을 유념해야 할 것이다.

#### 예 1: 사전행동적 인터페이스 - 일반

바람직하게, 본 발명의 사전행동적 사용자 인터페이스는, 사용자의 행동에 따라, 연산장치가 선택할 수 있는 다양한 미리 프로그램된 패턴 및/또는 특정한 사용자와의 이전 경험에 기초하여, 사용자에게 제안들을 능동적으로 하기 위해, 어떠한 타입의 연산장치도 제어할 수 있고 및/또는 그 장치와도 연관될 수 있는 것이 바람직하다. 상술한 제안들은 표시부의 적어도 일부의 모습을 변경함으로써, 예를 들어 메뉴 혹은 그 일부를 변경하고; 표시하기 위한 서로 다른 메뉴들을 제공하고; 및/또는 터치스크린 기능을 변경함으로써, 임의로 행해질 수 있다. 그 제안들은 음성에 의해 임의로 행해질 수도 있다.

사전행동적 사용자 인터페이스는 운영체제를 포함하는 상술한 바와 같은 연산장치를 위해 구현되는 것이 바람직하다. 임의로 및 바람직하게, 상기 인터페이스는 사용자와 운영체제와 사이의 통신을 위한 사용자 인터페이스를 포함하고 있다. 바람직하게, 인터페이스는 사용자의 사용자 인터페이스와의 상호작용의 적어도 하나의 패턴을 예를 들어 학습 모듈의 동작을 통해 검출할 수 있으며, 따라서, 바람직하게, 검출된 패턴에 따라 사용자의 인터페이스의 적어도 하나의 기능을 사전행동적으로 변경할 수 있게 된다. 따라서, 사전행동적 사용자 인터페이스는 사용자의 요청을 예상할 수 있고, 이에 따라 연산장치의 원하는 기능을 선택하는데 있어서 사용자에게 도움을 줄 수 있다.

특히 사용자가 원하는 것들과 그 행동을 학습하는 것과 관련하여, 이러한 타입의 사전행동적 행동은 사전행동적 인터페이스의 부분에 소정 타입의 학습능력을 필요로 하게 된다. 그러한 학습능력은 소프트웨어 객체들의 환경과의 상호작용들 및 (소프트웨어에 의한) 학습과 관련되는, 해당 기술분야에서 공지된 알고리즘과 방법론을 통해 임의로 제공될 수 있다. 소프트웨어가 그 행위들을 시간에 따라 향상시킬 수 있을 때 그 소프트웨어를 학습이라 말할 수 있다. 인공 지능은 지능형 행위 선택(추론, reasoning)을 보여줄 필요가 있으며, 이에 따라 소프트웨어가 그 환경(그 "세계")을 탐험하여 행위의 가능성들을 발견할 수 있는 능력을 갖는 것이 바람직하다. 또한, 소프트웨어는 세계의 상태와 그 자체의 내부 상태를 표현하는 능력을 갖는 것이 바람직하다. 그리고, 소프트웨어는 지능형 행위를 (상술한 지식을 이용하여) 선택하고 행동할 수 있는 것이 바람직하다.

임의로 및 바람직하게, 예를 들어 인터페이스의 학습모듈에 의한 학습은 보상들(rewards)에 의해 보강될 수 있고, 이때 학습 모듈은 환경의 상태에 따라 특정한 행위들을 취하는 것에 대해서 보상을 받게 된다. 이러한 형태의 학습은 소정의 방식으로 행동하도록 학습모듈을 훈련시키는 것을 포함한다. 만약 둘 이상의 행동이 허용되면, 학습처리과정은 비결정적이 되어 서로 다른 행동들을 생성할 수 있게 된다. 사전행동적 사용자 인터페이스와 관련하여, 임의로 및 바람직하게 상술한 보상은, 예를 들어, 제시된 선택 대상이 사용자의 선택으로 이어질 때(제시된 선택 대상으로 인해, 예를 들어 사전행동적 사용자 인터페이스에 의해 제공된 것과 다른 메뉴를 선택함으로써, 사용자가 다른 세트의 하나 이상의 선택들을 찾고자 하게 될 때와 반대)를 학습모듈이 검출하도록 유발하는 것 등을 포함된다. 사전행동적 사용자 인터페이스가 사용자 행동을 바로 이해하였음을 보여주는 것으로서, 분명히, 상기 인터페이스는 그 제시 사항으로 직접적인 사용자의 선택으로 이어지는 제시사항들의 퍼센트가 최대가 되도록 하려고 해야 한다.

이러한 처리에 도움을 주기 위해, 학습모듈에 의한 학습은 예를 들어 다음과 같은 임의의 그러나 바람직한 보강학습의 핵심 특징들에 따라 보강되는 것이 바람직하다:

**적응형 학습처리** - 학습처리는 반복적이어서, 각각의 반복에 대해서, 학습모듈은 수행할 적절한 행위를 학습하게 된다. 환경에서의 변화는 행동에서의 변화로 이어진다. 학습모듈은 특정한 행위들을 수행하도록 훈련되어질 수 있다.

**낮은 메모리소비** - 신경망 또는 MAS 등의 추론 시스템 알고리즘은 환경 상태 및 내부 상태가 작은 세트의 수치들로 출력될 때 때문에 메모리 복잡도가 낮다. 그러한 알고리즘은 학습처리 동안에 더 많은 메모리를 필요로 하지는 않는다.

**빠른 상호작용** - 임의로, 각각의 반복에 있어서, 이전의 반복에서의 연산에 기초하여 행위가 선택되어, 다음의 행위를 선택하는데 연산이 거의 수행되지 않게 된다. 사용자는 반응이 빠른 프로그램을 경험하게 된다. 다음 반복을 위한 학습처리는 그 행위가 일어난 후에 행해진다.

**장치 유휴 시간(device idle time)** - 학습 모듈은, 임의로, 사용자와의 상호작용이 없이 환경으로부터 스스로 학습할 수 있기 때문에, 유휴 연산 장치 시간이 학습을 위해 바람직하게 활용될 수 있게 된다.

도 1은 반응적 학습(reactive learning)을 위한 본 발명에 따른 학습 모듈의 예를 나타낸 개략 블록도이다. 도시된 바와 같이, 학습 모듈(100)은 지식베이스(102)를 바람직하게 포함하고 있으며, 지식베이스(102)는 환경과의 상호 작용의 결과로서 학습모듈(100)에 의해 수집된 정보를 보유함으로써 학습모듈(100)의 메모리로서 역할을 바람직하게 한다. 지식베이스(102)는 비휘발성 메모리(도시하지 않음)에 임의로 저장될 수 있다. 지식베이스(102)는 적절한 행위를 선택하기 위해 학습 모듈(100)에 도움을 주는 정보를 바람직하게 유지하고 있다. 그러한 정보는 내부 신경망을 위한 수치적 가중치(numerical weights) 등의 값들, 혹은 행위 보상값들을 갖는 테이블, 혹은 다른 형태의 정보를 임의로 포함할 수 있다.

학습모듈(100)이 환경에 대한 정보를 제공받을 수 있기 위해서는, 학습모듈(100)은 복수의 센서(104)를 특징적 요소로서 갖추는 것이 바람직하다. 센서들(104)은 학습모듈(100)이 그 환경상태를 인지할 수 있도록 한다. 센서들(104)은 그 환경에 연결되며 감지된 값들을 출력한다. 그 값들은 프로그램 자체로부터 얻어진 것(예를 들어, 화면상의 위치, 에너지레벨 등), 혹은 실제 장치 값들로부터 얻어진 것(예를 들어, "플리퍼(flipper)"를 열어서 그 장치가 작동되거나 착신호가 응답될 수 있게 되는 휴대폰을 펼친 상태(flipped state)와 같은 동작 상태 및 배터리 상태 등)일 수 있다.

센서(104)는 분명히 가치 있는 정보를 제공하지만, 학습모듈(100)이 그 정보를 이해하게 될 수 있기 전까지 그 정보를 처리할 필요가 있다. 따라서, 학습모듈(100)은 센서(104)의 현재 출력이 "상태(state)"라고 칭하는 그 세계의 일정한 표현이 되도록 처리하기 위해서 인지부(106)를 추가로 포함한다. 그리고, 그러한 상태는 학습모듈(100)의 "뇌(brain)"로서 기술될 수 있는 추론시스템(108)을 위한 입력이 되는 것이 바람직하다. 이러한 설계는 그 시스템을 여러 개의 호스트 플랫폼들(서로 다른 연산장치들 및 환경들)로 용이하게 이동하는 것을 지원할 뿐만 아니라, 센서의 메커니즘 및 세계의 상태를 확장하는 것을 지원하여, 세계의 상태가 장치에 따라 변경될 수 있게 된다.

추론시스템(108)은 지식베이스(102)로 현재 상태를 처리하여 어느 행위를 수행해야하는지 여부에 대한 의사결정을 생성하도록 하는 것이 바람직하다. 추론시스템(108)은 세계의 현재 상태를 수신하여, 수행되어야 할 행위를 출력하고, 그 선택된 행위에 대한 피드백을 수신하게 된다. 이 피드백에 기초하여, 추론시스템(108)은 지식 베이스(102)를 업데이트하는 것이 바람직하다. 이것은 학습모듈(100)이 행위들을 상태들과 연관시키도록 학습하게 되는 하나의 반복처리에 해당한다.

본 발명의 임의의 실시예에 의하면, 연산장치는 감정적 상태, 물리적 상태, 움직임 등의 사용자에 대한 여러 가지 형태의 생물학 정보를 감지하기 위한 하나 이상의 생물학 센서를 특징적 요소로서 가질 수 있다. 그리고, 그러한 정보는 센서(104)들에 제공되어, 인지부(106)가 사용자의 상태를 결정하고 그에 따라 그 장치를 위한 적당한 상태를 결정하는데 도움을 줄 수 있다. 그러한 생물학 센서는 체온, 맥박, 산소 포화도 감지용 센서 또는 사용자의 생물학 변수들을 측정하는 다른 모든 형태의 센서들을 포함할 수 있으나, 그것들에 한정되는 것은 아니다.

도 2는 학습 모듈(100)을 다시 특징으로 구비한, 사전행동적 사용자 인터페이스를 제공하기 위한 본 발명에 따른 시스템(200)의 전형적인 실시예를 나타낸 것이다. 학습 모듈(100)은 학습모듈(100)과 연관되고 및/또는 이것에 의해 제어되고 및/또는 이것을 동작시키는 연산장치(도시하지 않음)의 운영체제(202)와 통신을 하는 것이 도시되어 있다. 운영체제(202)는 사용자 인터페이스(204) 뿐만 아니라 적어도 하나의 다른 소프트웨어 애플리케이션(206)의 동작을 제어하는 것이 바람직하다(물론 그러한 많은 소프트웨어 애플리케이션들이 임의로 있을 수 있다).

사용자는 예를 들어 메뉴로부터 하나의 선택 대상을 선택함으로써 사용자 인터페이스(204)를 통해 통신을 한다. 운영체제(202)는 이러한 통신정보가 수신되어 데이터로 번역될 수 있도록 한다. 그리고, 바람직하게 학습모듈(100)은 그러한 데이터를 수신하여, 임의로 명령을 운영체제(202)로 다시 전송하여, 예를 들어 사용자 인터페이스(204)의 소정 요소를 (예를 들어, 다른 메뉴를 제공함으로써) 변경하고 및/또는 소프트웨어 애플리케이션(206)을 작동시킨다. 그리고, 사용자는 사용자 인터페이스(204)를 통해 응답을 하고; 이 응답으로부터, 바람직하게, 학습모듈(100)은 그 행위(학습모듈(100)에 의해 전송된 명령)가 적절한지 여부를 학습하게 된다.

도 3은 본 발명에 따른 사전행동적 사용자 인터페이스 시스템(300)의 구현 예를 나타낸 개략 블록도이다. 도시된 바와 같이, 시스템(300)은 바람직하게 3레벨 구조를 특징적 요소로서 가지며, 이때 애플리케이션 레이어는 인공지능(AI) 프레임워크에 의해 지원되며, 이것은 이어서 호스트 플랫폼 연산장치("호스트 플랫폼"으로 도시됨)와 통신을 하게 된다.

임의로 및 바람직하게, 애플리케이션 레이어는 복수의 서로 다른 애플리케이션들을 특징적 요소로서 가지고 있으며, 그 중에서, MutateApp(돌연변이 애플리케이션)(302), PreviousApp(이전 애플리케이션)(304) and TeachingApp(교육 애플리케이션)(306)와 같은 몇 가지 비한정적 예들을 도시하였다.

바람직하게, MutateApp(302)은 시스템(300) 내의 돌연변이들을 제어 및/또는 시작시키기 위해 호출(invocation)된다. 위에서 언급된 바와 같이, 학습모듈은 지시된 또는 반지시된 진화(directed or semi-directed evolution)를 통해, 예를 들어 유전적 알고리즘들을 통해, 그 행동을 임의로 변경할 수 있다. 바람직하게, MutateApp(302)는 진화를 통해 그러한 돌연변이들을 제어하고 및/또는 시작시킨다. 진화의 실시예에 대해서는 아래에 더욱 상세히 설명할 것이다.

바람직하게, PreviousApp(304)는 시스템(300)의 이전 상태 또는 (학습모듈의 상태와 같은) 그 일부분이 현재의 상태 대신에 호출될 수 있도록 한다. 좀더 구체적으로, 본 발명이 진화적 알고리즘으로 구현되는 것이라면 PreviousApp(304)는 사용자가 이전의 진화적 단계로 복귀하게 되도록 한다. 좀더 일반적으로, 시스템(300)은 상태 보존형(stateful)이므로, 이전 상태로 임의로 복귀할 수 있고, 이때 그러한 상태들의 히스토리는 바람직하게 유지되게 된다.

TeachingApp(306)은 예 3의 설명 이후에 더욱 상세히 설명되겠지만, 연산장치를 동작시키는 방법에 대해서, 및/또는 그 연산장치 외부의 다른 대상에 대해서 사용자에게 가르치도록 임의로 구현될 수 있다. TeachingApp(306)은

후술하는 인공지능 기반구조와 함께 개인 맞춤화된 학습 경험을 제공하는 교육 애플리케이션을 제공한다. 바람직하게, TeachingApp(306)은 교육 타입, 교육 방법들, 새로운 정보를 전달하는 비율(rate), 보강 활동들(reinforcement activities) 및 연습 활동들(practice activites) 등을 특정 사용자의 개별 요구들에 맞게 조정하는 것이 가능하게 되어 있다. 또한, 임의로 TeachingApp(306)은 예를 들어 그룹 혹은 교실 학습 상황에서 서로 다른 복수의 사용자들을 위해 성능을 조정할 수도 있게 되어 있다.

TeachingApp(306)은 인공 지능 프레임워크 레이어를 통해 구현될 수 있는 포괄적(generic) 애플리케이션의 하나의 비한정적 예일 뿐이다.

인공지능 프레임워크 레이어 자체는 사용자 인터페이스가 사전행동적 방식으로 행동할 수 있도록 하는 하나 이상의 구성성분을 포함하고 있다. 임의로 및 바람직하게, 프레임워크는 연산장치의 상태와 가상세계의 상태를 결정하고, 또한 그 두 상태 사이의 관계도 결정하기 위한 DeviceWorldMapper(장치 세계 매퍼)(308)를 포함한다. 바람직하게, DeviceWorldMapper(308)는 가상세계의 상태와 그 장치의 상태를 결정하기 위해, 예를 들어 EventHandler(이벤트 핸들러)(310)로부터의 다양한 이벤트로부터 입력을 받는다.

바람직하게, DeviceWorldMapper(308)는 입력데이터를 분석하기 위하여 AI(인공지능)/ML(Machine Learning)(312)와 통신을 하기도 한다. 바람직하게, 또한 AI/ML 모듈(312)은 여러 가지 자극에 대응하여 시스템(300)의 행동을 결정하며, 또한 예를 들어 사용자 인터페이스의 서로 다른 형태의 행위들에 대한 사용자의 반응으로부터, 시스템(300)이 학습을 할 수 있도록 한다. 시스템(300)의 행동은 진화 모듈(314)에 따라 임의로 및 바람직하게 향상될 수도 있다.

그러한 진화의 실시예는 이동 정보 장치 상의 지능형 에이전트의 사용과 관련되는 것이 특히 바람직하지만(예를 들어 이하의 설명 참조), 연산장치를 위한 어떠한 사전행동적 사용자 인터페이스와도 임의로 사용할 수 있다. 바람직하게, 이 실시예는 사전행동적 사용자 인터페이스가 아바타도 특징적 요소로서 가질 경우 또는 아바타와 결합되어 사용되는 경우에 이용되어진다.

바람직하게, 진화는 한 세트의 유전적 알고리즘들에 의해 모의(simulation)된다. 이들 알고리즘은 유전자, 염색체 및 표현형에 관해서, 사전행동적 인터페이스의 특성들(과 특히 아바타의 모습)을 기술하는 것에 기초하고 있다. 유전자는 예를 들어 특정 형태의 다리 등의 발현 수준(a level of expression)을 갖는 분별성(discrete property)이 있다. 발현 수준은 그러한 다리들의 개수일 수 있다.

표현형은 유전자의 외적 표현에 해당하며, 예를 들어 다리 유전자는 다리 길이 또는 다리 크기에 있어서, 다른 표현형을 가질 수 있다.

유전자는 돌연변이 과정을 임의로 거칠 수 있다. 이 과정(바람직하게는 특정 확률(probability)에 따른 과정)에 의해 유전자의 하나 이상의 변수(parameter)가 변경되어 다른 새로운 표현형들이 생성된다.

염색체는 함께 기능을 하는 한 세트의 유전자들에 해당한다. 염색체는 다른 생물로부터의 동일한 형태의 염색체와 혼성(hybrid) 또는 이종교배(cross breeding)되어 그 유전자의 부모 염색체들의 조합으로서 새로운 염색체를 생성하게 된다.

이 방법론은 사전행동적 사용자 인터페이스의 행위의 진화 및/또는 시각적 진화(예를 들어 아바타의 모습)를 모의하기 위해 포괄적 기반구조(generic infrastructure)를 생성하는데 도움을 준다. 이들 알고리즘은 영리함 및 체력 등의 비시각적 행동 특성을 위해서 임의로 이용될 수도 있다. 이러한 효과에 의한 결과로서, 예를 들어 움직임이 빠른 생물이 될 수 있고, 또는 더욱 능률적인 생물이 될 수 있다. 이들 알고리즘은 상술한 유전자/유전형/표현형의 구조에 따라 기술될 수 있는 그런 특성들 중 어느 것을 위해서도 임의로 이용될 수 있고, 이에 따라 예를 들어 행동적 유전자(behavioral gene)는 본 발명에 의해 이용되는 인공지능(AI) 알고리즘의 행동을 결정하게 된다.

바람직하게, 이러한 알고리즘의 출력은 여러 가지 가능한 자손 아바타들 및/또는 사전행동적 사용자 인터페이스들을 제공하게 된다.

유전적 알고리즘은 유전적 자식들 중 어느 것이 다음 세대로서 계속되게 할 것인지를 결정하기 위해 자연 선택 과정을 사용한다. 이러한 선택과정은 사용자에 의해 결정될 수 있고 혹은 미리 결정되어질 수도 있다. 이러한 방식으로 생물이 흥미로운 진화 행동을 표시할 수 있게 된다. 유전적 알고리즘 구성(framework)은 목표들 또는 성격(goals or character)과 같은 생물의 그 이외의 비시각적 특성들을 부호화(encode)하는 유전자들을 진화시키기 위해 이용될 수 있다.

진화모듈(314)은 예를 들어 MutateApp(302)의 동작을 통해 그러한 진화를 지원하고, 또한 바람직하게 그것을 관리하게 된다.

이러한 서로 다른 인공지능 타입의 애플리케이션들과 EventHandler(310)의 사이에서, 하나 이상의 서로 다른 저수준 관리자들(low level managers)은 서로 다른 이벤트들의 수신 및 취급(처리)과 시스템(300)에 의한 서로 다른 행동들을 바람직하게 지원하게 된다. 이들 관리자는 ActionManager(행위 관리자)(316), UIManager(UI 관리자)(318), StorageManager(스토리지 관리자)(320), 및 ApplicationManager(애플리케이션 관리자)(322)를 포함할 수 있으나, 이것들에 한정되는 것은 아니다.

ActionManager(316)는 이후 더욱 상세히 설명하겠으나, 간략히 설명하면, 예를 들어 AI/ML 모듈(312)의 동작을 통해서, 어느 행위를 취해야 하는지를 시스템(300)이 결정할 수 있도록 하는 것이 바람직하다.

바람직하게, UIManager(318)는 예를 들어 상술한 바와 같이 사용자 인터페이스에게 변화들을 지시하는 것에 의해, 사용자 인터페이스의 모습과 기능들을 관리한다.

바람직하게, StorageManager(320)는 예를 들어 시스템(300)의 지식베이스(도시하지 않음)와 관련한, 데이터의 저장 및 취급(처리)을 관리한다.

바람직하게, ApplicationManager(322)는 애플리케이션 레이어의 상술한 애플리케이션들과의 통신들을 취급한다.

바람직하게, 이들 서로 다른 관리자들은 모두 EventHandler(310)로부터 이벤트를 수신한다.

인공지능 프레임워크 레이어 내에서, 임의로 및 바람직하게, 인공지능 기반구조(324)는 호스트 플랫폼과의 통신을 지원한다. 바람직하게, 이 호스트 플랫폼 자체는 호스트 플랫폼 인터페이스(326)를 특징적 요소로서 가지며, 임의로 및 바람직하게, 이것은 예를 들어 호스트 플랫폼의 운영체제를 통해 제공될 수 있다.

임의로 및 바람직하게, 인공지능 기반구조(324)는 호스트 플랫폼 인터페이스(326)로부터 입력을 받고 또한 임의로 호스트 플랫폼 인터페이스(326)에게 명령들을 전송하기 위한 I/O 모듈(328)을 포함한다. 바람직하게, 스크린 모듈(330)은 호스트 플랫폼 연산장치의 화면상의 사용자 인터페이스의 표시를 처리/취급한다. 바람직하게, 자원 모듈(332)은 시스템(300)이 데이터 스토리지 등의 여러 가지 호스트 플랫폼 자원들(resources)에 접속할 수 있도록 한다.

물론, 상술한 도면들은 학습 모듈을 위한 단지 하나의 임의의 구성을 나타낼 뿐이다. 예를 들어, 학습모듈은 한 세트의 개별 에이전트들로서 표현될 수도 있으며, 이때 각 에이전트는 하나의 단순 목표를 가지고 있다. 학습모듈은 상술한 현재 상태에 기초하여 소정 행위를 수행하는 에이전트를 선택한다. 현재상태와 에이전트들 사이의 적절한 매핑은 보강 학습을 갖춘 학습모듈에 의해 학습될 수도 있다.

또한 학습은 임의로 감독될 수도 있다. 학습모듈은 한 세트의 행동 방법에 대한 예들을 보유하여 감독자로부터 부여된 패턴을 학습할 수 있게 된다. 학습모듈은 그러한 규칙들을 학습한 후에, 자신이 이미 알고 있는 정보에 기초하여 행동을 하고 새로운 상태들을 생성(또는 일반화)하려고 시도한다.

## 예 2: 이동 정보 장치를 위한 적응형 시스템

이 예는 이동 정보 장치를 갖춘 본 발명의 적응형 시스템의 예시적 구현에 관한 것이며, 이 구현은 바람직하지만 임의적인 것이고 어떠한 방식으로도 한정하도록 의도되는 것은 아니라는 점을 유념해야 한다.

이러한 적응형 시스템은 예 1에서 상술한 기능들 중 어느 것도 임의로 포함할 수 있고, 상술한 바와 같이 임의로 구현될 수도 있다. 이 절(section)에서는 이동 정보 장치의 동작과 관련하여 적응형 시스템의 실제 구조에 더욱 중점을 두기로 한다. 또한, 이 예는 본 발명에 따른 생물 혹은 아바타의 임의의 그러나 바람직한 구현에 대해서 기술하고 있다.

그 다음 절들은 본 발명에 따른 적응형 시스템의 다양한 측면들에 있어서의 특정한 기술적 구현들에 대한 임의의 그러나 바람직한 실시예들에 대해서 기술하고 있다. 단지 설명하기 위한 목적으로 또한 어떠한 한정할 의도 없이, 이들 실시예는 임의로 아바타 혹은 "생물"로서 시각적으로 표현되는 지능형 에이전트를 통해 사용자와 상호작용을 하는 적응형 시스템의 임의의 그러나 바람직한 실시예를 기초로 하고 있다.

### 제 1절: 이벤트 유도 시스템 (Event Driven System)

이 절에서는 본 발명에 따른 이벤트 유도 시스템의 바람직한 실시예에 대해서 기술하고 있으며, 이것에는 애플리케이션 관리자가 한정됨이 없이 포함되며, 그 장치에 의해 동작됨에 따라 본 발명의 시스템과 그 장치 자체와의 사이에서 일어나는 상호작용들에 대해서 기술하고 있다.

도 4는 본 발명에 따른 전형적인 적응형 시스템(400)과, 이 시스템(400)과 이동 정보 장치(402)와의 상호작용들을 나타낸 개략 블록도이다. 도시된 바와 같이, 바람직하게, 시스템(400)과 이동 정보 장치(402)는 모두 사용자(404)와 상호작용을 한다.

이동 정보 장치(402)는 많은 표준 기능들을 임의로 및 바람직하게 가지며, 도시된 바와 같이 그 기능들은 단지 설명하기 위한 목적으로 2 가지 카테고리(데이터 및 메커니즘들)로 분류된다. 메커니즘들에는 사용자 인터페이스(UI) 시스템(406)(스크린, 키패드 또는 터치스크린 입력부 등)과; 착신 및 발신 호 기능(408)과; 예를 들어 SMS를 위한 메시지 통신 기능(410)과; 착신 호 또는 메시지 및/또는 알람을 사용자(404)에게 알리기 위한 사운드부(412) 및/또는 진동부(414)와; 저장부(416) 등의 기능들이 임의로 포함될 수 있으며, 그러나 이것들에 한정되는 것은 아니다.

데이터에는, 주소록(전화번호부)(418); 착신 또는 발신 호 정보(420)와; 위치(422)와 같은 이동 정보 장치(402)의 위치와; 메시지 정보(424)와; 캐시 인터넷 데이터(426)와; 소유자 데이터(428)와 같은 사용자에 대한 데이터(404) 등이 임의로 포함될 수 있다.

이동 정보 장치(402)는 상술한 데이터/메커니즘들 중 어떠한 하나 이상도 임의로 포함할 수 있으며, 그들 모두를 반드시 포함하지 않아도 되고, 및/또는 도시되지 않은 부가적인 데이터/메커니즘들을 포함할 수도 있다. 이것들은 단순히 이동 정보 장치(402)와 관련한, 특히 휴대폰을 위한 비한정적 예들로서 의도된 것이다.

본 발명에 따른 적응형 시스템(400)은 적응형 (또한 바람직하게는 사전행동적) 사용자 인터페이스를 제공할 수 있기 위해 이동 정보 장치(402)의 데이터/메커니즘들과 상호작용을 함으로써, 사용자(404)가 이동 정보 장치(402)와 상호작용을 하는데 있어서의 용이함 및 효율을 증가시키게 된다.

바람직하게, 적응형 시스템(400)은 로직(430)을 특징적 요소로서 가지며, 바람직하게 이것은 상술한 학습 모듈과 유사한 방식으로 기능을 하게 되고, 또한 이것은 상술한 인공지능 및 기계 학습 알고리즘들에 따라 임의로 및 바람직하게 동작하게 된다.

로직(430)은 도 1(달리 설명되지 않는 한, 동일한 참조번호로 나타낸 구성요소는 기능이 동일하거나 유사함)을 참조하여 설명된 바와 같이 지식베이스(102)와 통신을 할 수 있는 것이 바람직하다. 바람직하게, 정보 저장부(432)는 이동 정보 장치(402)의 행위들에 대한 데이터와, 사용자 정보 등을 포함하며, 바람직하게는 지식베이스(102) 내의 데이터를 보충하게 된다.

바람직하게, 적응형 시스템(400)은 전화 로직(434)을 통해 전화할 수 있으며, 이것은 도 3의 MutateApp(302)과 전화 모듈(314)의 상술한 기능을 임의로 통합할 수도 있다(도시하지 않음).

임의로, 적응형 시스템(400)은, 언어 모듈(436)에 의해 지원되어, 텍스트 및/또는 가청 언어를 통해 사용자(404)와 직접 통신을 할 수 있다.

특히, 아래의 예 3에서 본 발명의 실시예에 관해 설명되어 있는 바와 같이, 그러나 또한 임의로 적응형 시스템(400)에 대해서, 사용자(404)는 사용자 인터페이스를 위한 아바타(도시하지 않음)를 임의로 제시받을 수 있다. 그러한 아바타는, 만일 이것이 있다면, 3D 그리픽 모델(438) 및 애니메이션 모듈(440)을 통해 임의로 생성될 수 있다(구체적인 설명은 아래의 설명참조). 임의로, 아바타는 사용자(404)를 위해 개별 맞춤화되어, 이동 정보 장치(402)와 상호작용할 때 사용자(404)를 위한 향상된 감정적 경험을 제공할 수 있다.

도 5a는 전형적인 애플리케이션 관리 시스템(500)의 개략 블록도로서, 이것은 본 발명의 적응형 시스템을 지원하기 위한 핵심적 기반구조에 해당한다. 상술한 바와 같이 또한 이후에 더욱 상세히 설명되는 바와 같이, 시스템(500)은 애플리케이션 기능을 교육시키는 것과 같은 실시예들을 지원하기 위해 임의로 이용될 수도 있다. 시스템(500)은 본 발명에 따른 적응형 시스템의 일부인 다른 형태의 애플리케이션들을 관리하기 위한 애플리케이션 관리자(502)를 특징적 요소로서 갖는 것이 바람직하다. 애플리케이션 관리자(502)는 BaseApp(베이스 애플리케이션)(504)라고 칭하는 애플리케이션 인터페이스와 통신을 하며, 이것은 시스템(500) 내의 모든 애플리케이션에 의해 구현된다. 애플리케이션 관리자(502)와 BaseApp(504)는 모두 EventHandler(506)를 통해 이벤트들을 교환(전달)한다.

애플리케이션 관리자(502)는 시스템 애플리케이션들(시스템(500)의 일부로서의 애플리케이션)의 실행을 위한 런타임을 관리하고 제공하는 역할을 한다. 그러한 애플리케이션들 각각의 라이프 사이클은 BaseApp(504)에서 정의되며, 이것에 의해 애플리케이션 관리자(502)는 그러한 애플리케이션들 각각을 시작, 일시정지(pause), 재개(resume) 및 종료(정지)시킬 수 있게 된다. 바람직하게, 애플리케이션 관리자(502)는 BaseApp(504)의 인터페이스의 스텝 방법(step method)을 통해 런타임 실행을 관리한다. 임의로 및 바람직하게, 상기 스텝 방법이 상태 보존형(stateful)인 것이 바람직하므로 시스템(500)은 실행을 위해 사용되어, 각 스텝이 하나 이상의 상태들에 (대략적으로) 대응하는 것이 바람직하다는 사실에 주목해야 한다. 하지만, 임의로, 실행(execution)은 모든 타입의 실행 방법 및/또는 스레드(threads)에 기초할 수도 있다.

애플리케이션 관리자(502)는 이동 정보 장치로부터 타이머 이벤트를 전달받는다. 아래에 더욱 상세히 기술하는 바와 같이, 바람직하게, 이동 정보 장치는 운영체제를 특징으로 가짐으로써 운영체제 레이어로부터 타이머 이벤트를 전달 받는 것이 바람직하다. 타이머가 호출(invoked)되면, 애플리케이션 관리자(502)는 수행되고 있는 현재 애플리케이션의 스텝을 호출한다. 사용자가 예를 들어 메뉴 시스템을 사용할 때 다른 애플리케이션을 구동시키면 애플리케이션 관리자(502)는 하나의 애플리케이션에서 다른 하나의 애플리케이션으로 전환되는 것이 바람직하다.

이어서, 상기 시스템 애플리케이션의 몇 가지 비한정적 예들을 나타내었으며, 여기에는 몇 가지 비한정적 예로서 도시된 바와 같이 TeachingMachineApp(교육 기계 애플리케이션)(508), MutateApp(돌연변이 애플리케이션)(510), GeneStudioApp(유전자 스튜디오 애플리케이션)(514), TWizardApp(티 마법사 애플리케이션)(516), FloatingAgentApp(부유 에이전트 애플리케이션)(518), TCWorldApp(티씨 월드 애플리케이션)(522) 및 HybridApp(혼성 애플리케이션)(520) 등이 포함되며, 이것에 한정되는 것은 아니다. 이들 애플리케이션에 대해서는 이후 도 3을 참조하여 더욱 상세히 설명하게 될 것이다.

바람직하게, MutateApp(510)는 적응형 시스템에 있어서의, 및/또는 사용자 인터페이스로서의 적응형 시스템을 나타내는 아바타의 모습에 있어서의, 돌연변이들을 제어 및/또는 시작시키기 위해 호출되어진다. 예 1을 이용하여 위에서 설명한 바와 같이, 본 발명의 적응형 시스템은 지시된 또는 반지시된 진화를 통해, 예를 들어 유전적 알고리즘을 통해, 그 행동을 임의로 변경할 수 있다. MutateApp(510)는 그러한 돌연변이들을 바람직하게 제어하고 및/또는 시작시키는 것이다.

바람직하게, GeneStudioApp(514)는 하나 이상의 수동 명령(manual commands)을 통하여 지시된 및/또는 반지시된 돌연변이들을 사용자가 실행할 수 있도록 한다. 예를 들면, 사용자는 적응형 시스템에게 특정한 입력을 받을 때 특정한 태스크 시퀀스(task sequence)를 수행하도록 (애플리케이션 관리시스템(500)을 통해) 지시하고자 할 수 있다. 혹은, 사용자는 만일 그것이 있다면 아바타의 모습 중 일부를 직접 변경하고자 할 수도 있을 것이다. 본 발명의 바람직한 실시예들에 따르면, 적응형 시스템의 그러한 서로 다른 형태들은 별개의 "유전자들"에 의해 바람직하게 구현되며, 이것은 사용자에 의해 임의로 변경되어 질 수 있다.

HybridApp(520)는, 사용자가 다른 이동 정보 장치의 적응형 시스템과 같은 외부 소스로부터 정보를 수신하여 그 정보를 사용자의 이동 정보 장치에 있는 기존 정보와 통합하고자 할 경우, 임의로 호출되어질 수 있다. 예를 들어, 사

용자는 다른 이동 정보 장치의 아바타와의 혼성된 모습을 갖는 아바타를 생성하고자 할 수 있다. 또한, HybridApp(520)는 임의로 및 바람직하게 아바타의 전체 진화적 상태에 대한 사용자의 주요 제어를 제공한다. 임의로 및 더욱 바람직하게, HybridApp(520)는 이름, 성격, 행동 및 모습을 임의로 가질 수 있는 아바타의 "생활(life)" 특성에 대해서 사용자에게 가르치는데 이용될 수 있다.

TeachingMachineApp(508)은 상기 장치 자체의 사용에 대한 명령(instruction)을 제공하는 것과 임의로 관련될 수 있는 애플리케이션의 예시적이고 비한정적인 예이며, 바람직하게는 장치 자체의 직접 동작과 관련하지 않는 대상에 대한 명령을 제공하다. 따라서, TeachingMachineApp(508)은 그 장치 자체의 사용 이외의 목적으로 이동 정보 장치 상에 제공되는 애플리케이션의 임의의 예(optional example)를 나타내는 것이다.

TCWorldApp(522)는 지능형 에이전트를 동작(run)시키는 애플리케이션이며, 바람직하게는 생물 혹은 아바타의 그 래픽 표시와 그 에이전트의 지능형 요소들(intelligent aspects)의 양쪽 모두를 제어한다(이 양쪽 모두에 대해서는 아래에 더욱 상세히 설명함).

본 발명에 따른 애플리케이션들에 대한 다른 비한정적 예들에 포함되는 것으로서 게임이 있다. 시스템 및 사용자가 함께 임의로 상호작용을 할 수 있는 게임의 한 가지 비한정적 예로서 "숨바꼭질" 게임이 있다. 바람직하게, "숨바꼭질" 게임은 메뉴 계층에 생물 또는 아바타를 "숨김"으로써 행해지게 되어, 바람직하게 사용자는 적어도 하나의 서브 메뉴를 가로질러 가서 아바타 혹은 생물을 찾음으로써 사용자가 메뉴 계층 및 구조에 대해서 더 많이 배울 수 있도록 한다. 그 이외의 그러한 여러 게임 애플리케이션들이 본 발명의 권리범위 이내에서 구현 가능하다.

TWizardApp(516)은 사용자에게 정보를 제공하는 다른 타입의 애플리케이션이다. 이것에 대해서는 아래의 예 4에서 시작 마법사 애플리케이션(Start Wizard application)과 관련하여 기술되어 있다. 간략하게 기술하면, 이 애플리케이션은 특히 감정형 시스템(이후 더욱 상세히 설명함)과 관련하여 또한 목표 우선순위(goal priorities)를 설정하는 것(이후 더욱 상세히 설명함)과 관련하여 지능형 에이전트의 특성과 같은 인공 지능 프레임워크의 구성 및 사용자 옵션(user preferences)을 포함하고 있다.

FloatingAgentApp(518)는 임의로 및 바람직하게, 특히 (만일 그것이 있다면) 아바타의 모습과 관련하여 사용자 인터페이스의 모습을 제어한다. FloatingAgentApp(518)는 사용자 인터페이스의 화상 표시의 요소들(aspects)이 아바타의 표시와 독립적으로 표시될 수 있도록 하며, 따라서 이것은 예를 들어 사용자 인터페이스 위로 "뜨게" 보일 수 있다. FloatingAgentApp(518)은 동작중인 다른 애플리케이션이 없을 때 구동되는 디폴트 애플리케이션인 것이 바람직하다.

도 5b는 본 발명에 따른 애플리케이션 관리자의 동작들을 나타내는 전형적인 시퀀스 다이어그램이다. 도시된 바와 같이, EventHandler(506)는 화살표 1로 나타낸 것처럼 애플리케이션 관리자(502)에게 이벤트에 대한 통지를 발송하는 것이 바람직하다. 그 이벤트가 타이머 이벤트일 경우에는, 애플리케이션 관리자(502)는 화살표 1.1.1.로 나타낸 바와 같이, 이미 호출된 관련 애플리케이션의 인스턴스의 스텝(행위)을 호출한다. 그 이벤트가 애플리케이션의 실행을 시작시키는 것이라면, 애플리케이션 관리자(502)는 화살표 1.2.1로 나타낸 바와 같이 관련 애플리케이션의 인스턴스를 호출한다. 애플리케이션의 현재 실행중인 인스턴스를 일시정지하고자 한다면, 애플리케이션 관리자(502)는 화살표 1.3.1로 나타낸 바와 같이 그 애플리케이션에게 일시정지 커맨드(pause command)를 전송한다. 애플리케이션의 이미 일시 정지된 인스턴스를 재개(resume)하고자 한다면, 애플리케이션 관리자(502)는 화살표 1.4.1로 나타낸 바와 같이 재개 커맨드를 그 애플리케이션에 전송한다. 어떠한 경우에도, 상술한 관련된 복귀 화살표로 나타낸 바와 같이, 그 스텝의 실행이 성공하면 애플리케이션 관리자(502)로 복귀되어진다. 그리고, 애플리케이션 관리자(502)는 EventHandler(506)에게 실행 성공(혹은 실패)에 대해서 통지한다.

이러한 서로 다른 애플리케이션들은 이동 정보 장치 동작의 여러 가지 요소들을 적응형 시스템이 제어할 수 있게 하는데 있어서 중요한 것이다. 하지만, 또한 적응형 시스템은 이동 정보 장치의 운영체제를 통해서, 여러 가지 이동 정보 장치의 구성요소들과 직접 통신하는 것이 가능할 필요가 있다. 그러한 통신은 도 6을 참조하여 나타낸 통신시스템(600)을 통해서 임의로 수행될 수 있고, 바람직하게는 행위 알고리즘들로 수행된다.

도 6a 및 도 6b는 각각 이동 정보 장치의 운영체제를 통하여 하나 이상의 행위를 수행하기 위해 본 발명에 따른 적응형 시스템에게 필요한 기반구조(infrastructure)의 예를 나타내는 도면(도 6a)과, 통신 시스템(600)의 동작을 위한 시퀀스 다이어그램의 예를 나타낸 도면(도 6b)이다. 본 발명의 임의의 그러나 바람직한 실시예들에 따르면, 이러한 기반구조는 "인공지능 랩퍼(AI wrappers)" 혹은 기존의 UI(사용자 인터페이스) 시스템을 획기적인 인공지능 및 기계 학습 기능들로 "포장(wrap)"하는 능력의 더욱 일반적인 개념의 일례이다.

통신 시스템(600)은 상술한 바와 같은 EventHandler(506)와 통신을 하는 베이스 클래스 이벤트(602)와 함께 다양한 형태의 이벤트들을 처리할 수 있는 것이 바람직하다. 그리고 EventDispatcher(604)는 본 발명의 시스템 내의 올바른 목표물(object)로 이벤트를 라우팅(전송)한다. 라우팅은 특정 이벤트에 대해서 EventDispatcher(604)로 그 목표물을 등록하여 결정되는 것이 바람직하다. EventDispatcher(604)는 그러한 통지를 위하여 EventHandler(506) 인터페이스를 구현하는 핸들러(handler)들의 레지스트리를 관리하는 것이 바람직하다.

임의로 및 바람직하게, 특별한 핸들러들이 구현되는 특정 이벤트들은 "플리퍼(flipper)"를 열어서 그 장치가 작동되거나 착신호(incoming call)가 응답될 수 있게 되는 휴대폰을 위한 플리퍼 이벤트 핸들러(606)를 포함하며, 플리퍼가 열리거나 닫히면 이 이벤트가 발생한다. 본 발명에 따라 동작되는 애플리케이션들은 이벤트들을 서로에게 임의로 보낼 수 있고, 그것들은 InterAppEvent 핸들러(608)에 의해 처리되는 것이 바람직하다. 생물 혹은 아바타의 임의의 그러나 바람직한 진화(변화)와 관련된 이벤트는 EvolutionEvent 핸들러(610)에 의해 처리되는 것이 바람직하다. 착신 혹은 발신 전화는 CallEvent handler(612)에 의해 처리되는 것이 바람직하며, 이것은 다시 2개의 추가 핸들러와, 통화를 개시하기 위한 CallStartedEvent 핸들러(614)와, 통화를 종료하기 위한 CallEndedEvent 핸들러(616)를 포함하는 것이 바람직하다.

SMS 이벤트(착신 혹은 발신 메시지)는 SMSEvent 핸들러(618)에 의해 처리되는 것이 바람직하다. 그 이벤트에 포함될 수 있는 임의의 그러나 바람직한 파라미터들은 이후 더욱 상세히 설명하는 바와 같이, 하나의 이동 정보 장치의 생물 혹은 아바타와 다른 이동 정보 장치의 생물 혹은 아바타와의 혼성화(hybridization)와 관련된 파라미터들을 포함한다.

그 키(keys)의 동작에 관련된 이벤트들은 KeyEvent 핸들러(620) 및/또는 KeyCodeEvent 핸들러(622)에 의해 처리되는 것이 바람직하다. 예를 들어, 사용자가 이동 정보 장치 상에서 키를 누르면, KeyEvent 핸들러(620)는 이 이벤트를 처리하며, 이것은 본 발명에 따른 시스템의 동작을 위한 착신 정보(incoming call)와 관련되는 것이다. 그 시퀀스 다이어그램에서, key\_event는 클래스 KeyEvent로부터의 객체이며, 이것은 키 이벤트 메시지 객체를 나타내는 것이다. KeyEvent 핸들러(620)는 key\_event 자체를 처리하는 반면 KeyCodeEvent 핸들러(622)는 입력 코드에 대해 주위를 기울인다(이 두 가지 입력 이벤트는 그 운영체제에 삽입되는 후크(hook)를 통해 얻어진다.).

BatteryEvent 핸들러(624)는 배터리 부족(low battery), 혹은 그 대신에 저 소비전력 모드에서 고 소비전력 모드로의 전환하는 것과 같은 배터리와 관련한 이벤트들을 처리하는 것이 바람직하다.

DayTimeEvent 핸들러(626)는 알람, 캘린더 혹은 리마인더/비지니스 다이어리(reminder/appointment diary) 이벤트들과 관련되는 것이 바람직하다.

도 6b는 이동 정보 장치 운영체제 혹은 다른 제어 구조와 본 발명의 시스템 사이에서 이벤트들이 어떻게 처리되는지를 나타내는 전형적인 시퀀스 다이어그램이다. 이 예에서, 상기 이동 정보 장치는 운영체제를 포함하고 있지만, 그러한 운영체제를 갖추지 못한 장치들을 위해 그와 유사한 동작 플로우가 임의로 구현될 수 있다. 만약 그것이 있다면, 운영체제는 상기 장치에 대한 입출력을 처리하며, 그 장치를 위해 발생하는 상태 및 이벤트들을 관리한다. 도 6b에서의 시퀀스 다이어그램은 그러한 이벤트들에 대한 (그리고 그들과 관련된) 처리를 용이하게 하기 위해 추상화(abstraction)한 것이다.

운영체제 모듈(os\_module)(628)은 이벤트를 발생시키거나 그와 관련된 것으로, 임의로 그러한 모듈이 복수 개 존재할 수 있으며, 명확히 하기 위해 도면에는 하나만 나타내었지만 이것은 어떠한 방식으로도 한정하고자 하는 것은 아니다. 운영체제 모듈(628)은 이동 정보 장치의 운영체제의 일부이다. 운영체제 모듈(628)은 운영체제 모듈(628)에 의해 수신되었는지 생성되었는지 여부의 이벤트에 대한 통지를 후크(630)에 전송하는 것이 바람직하다. 후크(630)는 본 발명에 따른 시스템의 일부로서, 본 발명에 따른 시스템과 운영체제 사이의 통신을 허용하기 위해 사용된다. 후크(630)는 운영체제로부터의 이벤트를 해석할 수 있고, 이벤트(602)에게 이해될 수 있는 메시지 내에 이벤트를 구성할 수 있다. 후크(630)는 EventDispatcher(604)에 그 이벤트를 발송하기도 하며, 이것은 (그러한 핸들러가 복수 개 존재할 수 있으나) EventHandler(506)로 도시한 그 이벤트를 위한 각 핸들러와 통신을 한다. 그리고 EventDispatcher(604)는 후크(630)에 보고하며, 이것은 그 이벤트의 처리에 대해서 운영체제 모듈(628)에게 보고한다.

도 7a 내지 도 7c는 전형적인 이벤트들을 나타낸 것으로, 그 이벤트들이 (이동 정보 장치의 운영체제를 통해) 이동 정보 장치와 본 발명의 시스템과 사이의 상호작용들에 의해 어떠한 방식으로 처리되는지를 나타내 도면이다. 이동 정보 장치와 연관하지 않으면서, 몇 가지 이벤트들이 본 발명의 시스템 내에서 임의로 처리될 수 있다는 점을 유념해야 한다.

도 7a는 퀄컴사의 MSM (Messaging State Machine) CDMA (Code Division Multiple Access) 이동 플랫폼을 위한 퀄컴사의 DMSS 운영체제 기반구조를 갖는 이동 정보 장치에 따라 기술된 전형적인 키 이벤트 시퀀스 다이어그램을 나타낸 것이다. 이 운영체제는 전화키(키패드)를 사용하여 입력된 대화형 입력(interactive input), I/O 서비스 및 사용자 인터페이스 서비스와 같은 운영체제 서비스를 제공한다. 이 예는 본 발명의 시스템에 의해 키로부터의 입력 이벤트가 어떻게 생성되고 처리되는지를 보여주고 있다. 다른 이벤트들은 거의 동일한 방식으로 상기 시스템에 전송되지만, 후크(630)의 기능은 그 이벤트를 전송하고 있는 운영체제 모듈에 따라 변경되며, 임의로 및 바람직하게, 그러한 후크가 복수 개 존재하여 후크가 각각 운영체제와의 상호작용과 관련한 서로 다른 기능을 가지고 있다.

도 7a에 나타낸 바와 같이, ui\_do\_event 모듈(700)은 상기 운영체제의 구성요소로서 주기적으로 호출된다. 상기 이동 장치의 키가 눌려지면, 정보를 ui\_do\_event 모듈(700)에 전송하는 사용자 인터페이스(UI) 구조는 그 키의 값을 포함하게 된다. 그러면 후크(630)는 그 키 값을 수신하고, (특히 ui\_do\_event 모듈(700)이 글로벌 이벤트를 발송(dispatch)하면) 임의로 및 바람직하게 그 이벤트를 키 이벤트로서 식별하게 되고, 키 이벤트(702)를 생성한다. 이어서, 그 이벤트는 바람직하게는 상술한 바와 같은 이벤트 핸들러(도시하지 않음)를 통하여, 그러한 이벤트에 대한 통지의 수신을 요청한 애플리케이션(704)으로 전송된다. 이어서, 바람직하게, 그 이벤트의 처리 중에 성공(혹은 실패)의 통지가 EventDispatcher(604)로 다시 되돌아가게 되어, 후크(630)와 ui\_do\_event 모듈(700)로 전달되게 된다.

도 7b는 이벤트를 처리하기 위한 시퀀스 다이어그램의 제2 구체 예를 나타낸 것으로, 이 경우 그 이벤트는 본 발명에 따른 시스템으로부터 전달되고, 이동 정보 장치의 스크린 상의 그림(drawing)과 관련된다. 정보는 스크린이 (주로) 프레임 버퍼에 의해 표현되는 운영체제의 스크린 엑세스 방법을 통해 전달된다. 이 프레임 버퍼는 스크린 드라이버를 이용하여 카피되고 그 스크린에 의해 표시되는 메모리 세그먼트이다. 본 발명의 시스템은 운영체제에게 스크린 상의 그림을 제어하기 위해 필요한 정보를 발생시킨다.

도 7b를 다시 참조하면, 화살표 1로 나타낸 바와 같이, (scrn\_update\_main 모듈(710)을 통하여) 운영체제는 우선 스크린을 위하여 프레임 버퍼를 생성한다. 이 생성은 예를 들어 임의로 백그라운드를 그리는 것과 연관될 수 있으며, 이것은 본 발명의 시스템에 의해 제공된 정보로부터의 데이터가 그려지지 않은 스크린의 모든 부분에 표시될 수 있다. 임의로, 그러한 백그라운드가 있음으로써 반투명 윈도우(semi-transparent windows)의 사용을 지원하게 되며, 이것은 아래에 더욱 상세히 기술되는 바와 같이 생물 혹은 에이전트를 위해 임의로 및 바람직하게 이용될 수 있다.

그러면, Scrn\_update\_main 모듈(710)은 생성된 데이터를 위한 요청을 스크린 모듈(712)에게 전송하며, 이것은 본 발명의 시스템의 일부로서 운영체제와 통신하기 위한 후크를 특징으로 갖는다. 이어서 스크린 모듈(712)은 스크린에 그려져야 하는 것에 대한 생성된 정보를 위해서 각 애플리케이션 윈도우(도면에는 agentWindow(에이전트 윈도)(714)로 나타내었고 임의로 복수 개가 있을 수 있음)에게 요청을 전송한다. 변경이 발생하여 생성이 필요하면, agentWindow(714)는 생성이 필요하다는 것을 스크린 모듈(712)에게 통지한다. 그러면 스크린 모듈(712)은 변경된 부분의 위치 및 크기를 바람직하게 두 번의 별개의 요청(화살표 2.1.2.1 및 2.1.2.2로 각각 도시함)으로 요구하게 되며, 그것에 대해서는 응답이 agentWindow(714)에 의해 전송된다.

스크린 모듈(712)은 (바람직하게는 다음과 같이) 생성된 직사각형 형태로 그 정보를 scrn\_update\_main(710)을 통하여 운영체제에 반송한다. Scrn\_update\_main(710)은 프레임 버퍼를 프리 버퍼(pre-buffer)에 카피함으로써 생성이 있다는 것에 대한 통지에 응답한다(과정 3.1). 그러면 스크린 모듈(712)은 화살표 3.2.1로 나타낸 것처럼 프리 버퍼에 각 윈도우에 대한 변경들을 그려 넣는다. 이어서 프리 버퍼가 프레임 버퍼에 카피되고 이에 따라 스크린에 카피된다(화살표 3.3).

도 7c는 스크린상에 그리기 위한 본 발명의 시스템을 위한 클래스 아키텍처를 나타낸 것이다. 스크린 모듈(712) 및 agentWindow(에이전트 윈도)(714)를 모두 도시하였다. 그 클래스 agentWindow(714)도 세 개의 다른 윈도우 클래스들과 통신을 하며, 이것들은 BackScreenWindow(백스크린 윈도)(716), BufferedWindow(버퍼 윈도)(718) 및 DirectAccessWindow(직접 액세스 윈도)(720) 등의 윈도우의 생성(또는 변경)에 관한 정보를 제공한다. BufferedWindow(718)는 그 자체와 통신을 하는 TransbufferedWindow(트랜스 버퍼 윈도)(722) 및 PreBufferedWindow(프리 버퍼 윈도)(724) 등의 2개의 윈도우 클래스를 추가로 가진다.

## 제2 절: 행위 선택 시스템(Action Selection System)

이 절에서는 인센티브(들)/행동억제요인(들)에 따른 임의의 행위 선택 등을 포함하는 (그것에 한정되는 것은 아님) 본 발명에 따른 행위 선택 시스템의 바람직한 실시예에 대해서 기술하기로 한다. 지능형 에이전트의 행위가 어떻게 선택되는지에 대한 설명을 돋기 위해서, 본 발명의 시스템에 의해 바람직하게 제공되는 가상 환경과의 지능형 에이전트의 상호작용과 지능형 에이전트의 구조에 대해서 우선 설명하기로 한다.

도 8a는 지능형 에이전트의 구성 예를 나타낸 것이고, 도 8b는 그 지능형 에이전트의 동작에 대한 시퀀스 다이어그램의 예를 나타낸 도면이다. 도 8a에 나타낸 바와 같이, 지능형 에이전트(800)는 복수의 클래스를 포함하는 것이 바람직하다. 메인 클래스는 AICreature(인공지능 생물)(802)이고, 이것은 그 상태, 성격, 목표 등과 같은 지능형 에이전트에 대한 정보를 포함하며, 또한 위치, 색깔 및 현재 볼 수 있는지 여부 등의 그 에이전트를 시작적으로 나타내는 생물의 모습에 대한 정보를 포함한다.

AICreature(802)는 World(세계)(804)와 통신을 하며, 이것은 지능형 에이전트용 가상 환경을 위한 기초 클래스이다. 또한, World(804)는 가상 환경을 포함하는 클래스들과 통신을 하며, 그것에 대한 몇 가지 비한정적 예를 도시하였다. 바람직하게, World(804)는 WorldObject(세계 객체)(806)의 여러 가지 인스턴스들(instances)과 통신을 하며, WorldObject(806)는 가상 환경에서 발견되는 객체를 나타내며 지능형 에이전트와 상호작용을 할 수 있는 것이다. World(804)는 그러한 서로 다른 객체들을 관리하며 또한 그들에 대한 위치 등의 속성(property)을 포함하는 특성(characteristics)에 대한 정보를 수신한다. 또한 World(804)는 스스로 크기 및 가시성(visibility) 등의 가상 환경의 속성들을 관리한다. WorldObject(806)는 2차원 혹은 3차원 그래픽 혹은 그것들이 혼재된 것을 임의로 이용할 수 있고, 또한 소리 생성 등의 이동 정보 장치의 다른 기능들을 임의로 이용할 수도 있다.

WorldObject(806)는 스스로 몇 개의 클래스들 중 하나에 속하는 객체를 임의로 나타낼 수 있다. 이러한 추상화(abstraction)는 서로 다른 객체 클래스들이 가상 환경에 추가되거나 그것으로부터 제거될 수 있도록 한다. 예를 들어, 임의로 상기 객체는 MenuBallObject(메뉴 공 객체)(808)에 의해 표현된 바와 같이 예를 들어 메뉴의 일부로서 시작될 수 있는 "공"이 될 수 있고, 그 공을 갖고 놀기 위해 생물에 의해 "제거"될 수 있다. 바람직하게, GoodAnimalObject(좋은 동물 객체)(810)도 WorldObject(806)와 통신을 하며, 그리고 바람직하게 (생물을 위한 음식을 나타내는) FoodObject(음식 객체)(812)와, (생물을 화나게 하여 예를 들어 싸움을 하게 할 수 있는 동물을 나타내는) BadAnimalObject(나쁜 동물 객체)(814)와, (그 생물을 위한 집) HouseObject(집 객체)(816) 등의 클래스들은 GoodAnimalObject(810)와 통신을 한다. GoodAnimalObject(810)는 스크린 상의 객체 등을 그릴 수 있는 기능을 포함하며, 이 때문에 바람직하게 다른 클래스들 및 객체들이 GoodAnimalObject(810)와 통신을 하는 것이다. 물론, 예를 들어 그 생물에 다른 완구(toy)들이 제공될 수 있으므로 이 시스템에서는 다른 많은 클래스 및 객체들이 가능하다.

임의로 및 바람직하게, WorldObject(806)는 예를 들어 그 상태에 등급화된 입력(graded input)을 제공함으로써, 지능형 에이전트의 상태와 관련될 수도 있다. 바람직하게, 이 입력은 지능형 에이전트에게 인센티브를 제공하거나 행동억제요인을 제공한다는 의미에서 등급화되며, 또한 그것은 중성적 영향(neutral influence)을 줄 수도 있다. 바람직하게, 그러한 등급화된 복수개의 입력들을 집계(aggregation)함으로써 지능형 에이전트의 상태가 결정되게 된다. 도 8a의 시퀀스 다이어그램과 도 9a 및 도 9b 각각의 그래프 탐색 방법 및 행위 선택 방법 다이어그램을 참고하여 기술된 바와 같이, 바람직하게 가상 환경으로부터 지능형 에이전트로 부여되는 보상을 최대화하기 위하여 등급화된 입력들이 집계되어진다.

그러한 등급화된 입력들은 장려 혹은 방해 피드백(encouraging or discouraging feedback) 형태의 사용자로부터의 입력을 임의로 포함할 수도 있어, 지능형 에이전트는 피드백이 제공되어진 행동을 계속하는 것에 대한 인센티브 혹은 행동억제요인을 각각 갖게 된다. 사용자로부터의 피드백에 대한 세계 상태에 대한 계산을 임의로 및 바람직하게 다음과 같이 수행할 수 있다:

$$\text{Grade} = (\text{weighting\_factor} * \text{feedback\_reward}) + ((1 - \text{weighting\_factor}) * \text{world\_reward})$$

이 식에서, `feedback_reward`(피드백\_보상)는 사용자에 의해 제공된 피드백에 의한 것이고, `world_reward`(세계\_보상)는 상술한 바와 같은 가상 환경으로부터의 집계된 총 보상이며, `weighting_factor`(가중\_인수)는 바람직하게 및 임의로 0과 1 사이의 값으로 가상 환경(세계) 피드백과 반대되는 사용자 피드백의 가중치(weight)를 나타내는 것이다.

도 8b는 본 발명의 지능형 에이전트와 가상 세계 사이의 일련의 상호작용들의 예를 나타낸 전형적인 시퀀스 다이어그램이다. 이 시퀀스는 지능형 에이전트의 상태(status)에 대한 개선을 위해서 가상 세계 모듈(818)로부터 `AICreature`(812)로의 요청으로 시작된다. 가상 세계 모듈(818)은 스스로 지능형 에이전트를 포함하여 전체 가상 환경을 제어 및 관리한다.

그리고 지능형 에이전트는 화살표 1.1.1로 나타낸 바와 같이 행위를 수행할 것인지를 고려한다. 그 행위는 바람직하게 세계(804) 및 세계 객체(806)와 상호작용함으로써, 모든 세계 객체를 통하여, 이어서 순환적(recursive)으로 각 객체를 위한 모든 행위를 통하여 탐색(화살표 1.1.1.1)을 거쳐 선택된다. 각 행위에 대한 잠재적 보상이 구해지며(화살표 1.1.1.1.1), 등급화된다(화살표 1.1.1.1.1.2). 최고의 보상값을 갖는 행위가 선택된다. 그 다음으로 지능형 에이전트를 위한 전체적 등급이 결정되고 `AICreature`(802)는 선택된 행위를 수행한다.

이어서, `Virtual_world`(가상\_세계)(818)는 `World`(804) 및 `WorldObject`(806)와 통신을 함으로써 세계 안의 모든 객체들의 위치 및 상태를 개선한다.

여러 가지 잠재적 행위들에 대한 검색이 하나 이상의 다수의 서로 다른 방법에 따라 임의로 수행될 수 있다. 도 9a 및 도 9b는 본 발명에 따른 행위를 선택하기 위한 2개의 전형적인 방법을 나타낸 것이다.

도 9a는 행위를 선택하기 위한 규칙 기반 방식이라고 하는 행위 선택 방법의 예를 나타낸다. 단계 1에서, 가상 환경의 상태(status)가 `World state`(세계 상태)에 의해 결정된다. `World Event`(세계 이벤트)가 발생한 후, 그 이벤트에 적절한 `State Handler`(상태 핸들러)가 단계 2에서 호출된다. 상태 핸들러는 단계 3에서 지식 베이스에게 문의를 한다. 임의로, 이 지식 베이스는 호출된 상태 핸들러에 따라 별개의 셕션들 및/또는 별개의 지식 베이스들로 분할 될 수 있다. 단계 4에서, 응답이 상태 핸들러에게 전달된다.

단계 5에서, 규칙 베이스 유효성 판단(validation)이 수행되며, 이때 상기 응답(과 그에 따른 지능형 에이전트를 특정 상태가 되도록 하는 제안된 행위)가 그 규칙들과 비교된다. 그 행위가 유효하지 않다면, 처리는 단계 1로 복귀된다. 행위가 유효하다면, 단계 6에서 그 행위가 생성된다. 그 후 그 행위에 대한 우선순위(이후 도 9c를 참고로 하여 더욱 상세히 설명하기로 함)가 단계 7에서 바람직하게 결정되며, 더욱 바람직하게는 그 우선순위는 복수의 입력들에 따라 결정되며, 그 입력들에는 행위 확률(probability), 행위 유틸리티 및 사용자 성향(user preference)이 포함되며 이것에 한정되는 것은 아니다. 단계 8에서 그 행위는 행위 관리자를 위해 큐(대기 행렬)(queue)에 배치된다. 단계 9에서, 행위 관리자는 최고 우선순위 행위를 독출하고, 그 독출된 행위는 단계 10에서 지능형 에이전트에 의해 수행된다.

도 9b는 그래프 탐색 방법에 따른 전형적인 행위 선택 방법을 나타낸 것이다. 다시, 단계 1에서 그 처리가 세계의 상태(가상 환경)를 결정함으로써 시작하게 되며, 그 상태에는 그 세계 내에 있는 객체들의 상태 및 지능형 에이전트의 상태가 포함된다. 단계 2에서, 지능형 에이전트가 조회(query)된다. 단계 3에서, 지능형 에이전트는 각 세계 객체를 위한 일련의 (허용된 혹은 가능한) 법적 행위들을 얻게 되고, 바람직하게 각 세계 객체가 도시된 바와 같이 조회된다.

이제, 상기 방법은 2개 부분으로 갈라지게 된다. 첫 번째 부분에서는, 오른쪽에 나타낸 바와 같이, 각 행위 경로(action path)를 위해 수행된다. 단계 4에서, 수행될 행위가 모의(simulated)된다. 단계 5에서, 그 시뮬레이션의 효과(effect)가 그 세계를 위해 결정되고, 단계 6에서 바람직하게 각 세계 객체를 위해 결정된다. 단계 7에서, 각 행위의 효과를 위해 등급이 결정된다.

단계 8에서, 행위의 전체 축적된 보상과 같이, 객체들의 상태 및 그에 따른 세계의 상태가 결정된다. 단계 9에서, 행위의 효과가 지능형 에이전트에 대해서 모의되고, 바람직하게 단계 10에서 각 세계 객체와 지능형 에이전트 사이의 효과도 고려된다.

이제 상기 방법의 왼쪽 부분에 대해서 살펴보면, 단계 11에서, 바람직하게 이러한 정보가 모두 최고 보상을 갖는 행위 경로를 결정하는데 이용된다. 단계 12에서, 행위가 생성된다. 단계 13에서, 바람직하게 행위 등급 또는 보상에 따라 행위 우선순위가 설정된다. 단계 14에서, 도 9a의 경우와 같이 행위 관리자에서 큐에 배치된다. 단계 15에서, 그 행위가 우선순위에 따라 행위 관리자에 의해 고려되어지며, 우선순위가 가장 높은 행위가 선택되어져서 단계 16에서 바람직하게 수행되게 된다.

다음으로, 전형적인 행위 실행 방법 및 구조에 대해서 기술하기로 한다. 도 10은 본 발명에 따른 전형적인 행위 실행 방법의 시퀀스 다이어그램을 나타낸 것이다. 핸들러(1000)는 화살표 1로 나타낸 것처럼 행위 모듈(1002)에게 행위를 위한 목표(goal)를 전송하고, 모듈(1002)은 베이스 행위 인터페이스를 특징으로 갖는 것이 바람직하다. 이 베이스 행위 인터페이스는 행위 모듈(1002)로 하여금 핸들러(1000)와 통신을 하고 또한 시스템의 다른 객체들과 통신을 할 수 있도록 하며, 이것들은 여기에서는 `FloatingAgentApp`(부유 에이전트 애플리케이션)(1006)으로 나타낸 지능형 에이전트에 의한 추후 실행을 위해 행위들을 생성하고 배치(post)하는 것이 가능하다. 이를 행위는 행위 관리자(1004)에 의해 관리된다.

행위 관리자(1004)는 행위 객체들을 포함하는 2개의 큐를 가지고 있다. 그 중 하나의 큐는 실행 준비 완료상태 큐(ready for execution queue)이며, 다른 하나는 실행 계류상태 큐(pending for execution queue)이다. 두 번째(후자) 큐는 예를 들어 행위가 생성되었을 때 사용될 수 있으며, 그러나 그 행위의 내부 상태는 계류상태이어서 그 행위

는 실행을 위해 준비되지 않은 상태이다. 그 행위가 실행 준비가 완료되도록 성숙되어지면, 그 행위는 상기 실행 준비 완료상태 큐로 이동되는 것이 바람직하다.

애플리케이션 관리자(1008)는 화살표 2로 나타낸 바와 같이 행위를 실행하기 위해 FloatingAgentApp(1006)과 상호작용을 하는 것이 바람직하다. 그리고 나서 FloatingAgentApp(1006)는 행위 관리자(1004)에게 다음 행위를 요구하는 것이 바람직하며(화살표 2.1), 그 행위 자체는 행위 모듈(1002)에 의해 제공되는 것이 바람직하다(화살표 2.2.1). 바람직하게, 행위들은 핸들러(1000)로부터 행위 관리자(1004)로 대기상태(enqueued)가 된다(화살표 3). 바람직하게, 목표(goal) (및 그에 따른 우선순위의 적어도 일부분)은 핸들러(1000)와 행위 모듈(1002) 사이의 통신에 의해 각 행위를 위해 설정된다(화살표 4). 화살표 5 및 6은 아래에 더욱 상세히 설명하는 "harakiri()" 방법을 나타내고 있다.

전술한 바와 같이, 그 행위들은 우선순위 순서에 따라 대기상태(queued)가 되는 것이 바람직하다. 우선순위는 행위 관리자(1004)에 의해 행위 모듈(1002)의 인터페이스에 대한 조회를 통하여 결정되는 것이 바람직하다. 바람직하게, 행위의 우선순위는 복수의 파라미터들을 포함하는 계산에 따라 결정된다. 예를 들어, 바람직하게 이 파라미터들에는 상기한 생성되는 객체에 의해 도출 혹은 추론된 것으로서의 우선순위를 포함하며, 더욱 바람직하게 이것은 행위의 성공을 위한 예측된 확률(predicted probability)과; 그러한 타입의 행위에 의한 과거 경험에 따라 (예를 들어 사용자의 수요 및 행위 성공에 따라) 결정되는 것이 바람직한 그러한 타입의 행위의 지속적 우선순위(persistent priority)와; 사용자 성향(user preferences)에 따라 결정되는 것이 바람직한 목표 우선순위(goal priority) 등을 기초로 한다.

상기 파라미터들을 관리하기 위한 한 가지 임의의 계산은 다음과 같다:

$$P(\text{all}) = P(\text{action probability}) * ((P(\text{persistent priority}) + P(\text{action goal})/10))/2$$

우선권(우선순위)에 기초한 행위 실행에 대한 보완으로, 각 행위는 생존시간(Time To Live(ttl))이라는 시간을 참조 가능(referable)하게 가지며; 이 ttl 값은 행위가 상기 준비 완료상태 큐에 배치된 때와 이 행위의 종료시간 사이에 경과된 실행 시간의 양을 가리킨다. 행위가 준비 완료상태에 있지만 그 ttl이 종료된 때까지 실행 우선권(priority)을 전달받지 않은 경우, 행위 관리자(1004)는 상기 방법 harakiri()를 호출하는 것이 바람직하고, 이것은 그 행위에게 그것이 실행되지 않을 것이라는 것을 통지한다. 그러한 harakiri()의 호출은 각각 기준치(threshold)에 도달될 때까지 행위의 우선순위를 감소시키는 것이 바람직하다. 이 기준치에 도달된 후에는, 지속적 우선순위가 증가하기 시작한다. 이러한 모델은 제안 혹은 실행되었으나 사용자가 그 행위를 중단하여 실패된 행위들을 처리하도록 동작한다. 상기 지속적 우선순위는 행위 우선순위 계산에 과거 경험을 포함시킴으로써 감소하게 된다.

이러한 방법은 제안 혹은 실행된 행위들이 실행시에 특정 사용자의 비명시적인 성향에 어떠한 방식으로 적응되지는지를 보여주고 있다.

행위의 지속적 우선순위가 감소하여 그 행위가 동작하지 않을 경우에, 예를 들어 사용자가 변경을 선호하면, 그것이 제거되거나 혹은 (그밖에 가능하게는) 다시 동작하는 것이 허용될 필요가 있기 때문에, 이러한 모델은 harakiri() 메커니즘이 없으면 완전하지 못하다. harakiri()의 몇 가지 실행들 이후에, 그 행위가 동작하기 위한 우선순위를 다시 회복할 수 있다.

상술한 절들(Sections)에서는, 여러 가지 행위들 및 메커니즘이 본 발명의 적응형 시스템을 통해 실행될 수 있도록 하는 기반구조를 제시하고 있다. 이를 행위 및 메커니즘에 대해서 아래에 더욱 상세히 설명할 것이다.

### 제3 절: 감정형 시스템(Emotional System)

이 절에서는 본 발명에 따른 감정형 시스템의 바람직한 실시예를 기술하고 있으며, 특정한 감정들 및 그 감정들의 정도(intensities)에 대한 설명이 포함되어 있으나 이것에 한정되는 것은 아니며, 바람직하게 그것들은 서로 결합되어 전체적 기분(mood)을 형성한다. 또한, 바람직하게 상기 감정형 시스템은 기분이 변경되도록 하기 위한 것일 뿐만 아니라 예를 들어 변화율과 같은 그러한 변화의 하나 이상의 측면을 임의로 제어하기 위한 메커니즘을 포함한다.

도 11a 내지 도 11c는 본 발명에 따른 전형적인 감정형 시스템의 구현 예를 기술하기 위한 다이어그램들을 특징으로 하는 도면이다. 도 11a는 본 발명에 따른 감정형 시스템을 위한 전형적인 클래스 다이어그램을 나타낸 도면이고, 반면에 도 11b 및 도 11c는 본 발명에 따른 감정형 시스템의 동작을 위한 전형적인 시퀀스 다이어그램을 나타낸 도면이다.

본 발명에 따른 감정형 시스템(1100)에 대해서 나타낸 바와 같이, 목표 클래스(goal 1102)는 지능형 에이전트의 추상적 목표(abstract goal)를 나타낸다. "목표(goal)"란 지능형 에이전트가 행위를 수행하여 성취하고자 하는 무엇을 말한다. Goal(1102)은 그 목표의 상태와 그 이행 가능성(chances of fulfillment)과 관련된 소정의 이벤트들에 기초하여 감정들을 생성하는 역할을 한다.

Goal(1102)은 AICreature(802)와 상호작용을 한다(도 8을 참고로 상술함). 이들 상호간섭들에 대해서는 아래에 더욱 상세히 기술되어 있다. 요약하여 설명하면, 인공지능 에이전트는 목표들을 이행하려고 하여, 목표들이 이행되었는지 여부를 판단하기 위해 AICreature(802) 상이의 상호작용들이 필요하며, 이것은 다시 지능형 에이전트의 감정적 상태에 영향을 주게 된다.

감정적 상태 자체는 클래스 EmotionalState(1104)에 의해 처리되며, 그리고 이것은 클래스 Emotion(1106)과 연결되어 있다. Emotion(1106) 자체는 노함 클래스 AngerEmotion(1108) 및 즐거움 클래스 JoyEmotion(1110)과 같은 특정 감정들을 위한 클래스들과 연결되어 있는 것이 바람직하다. EmotionalState(1104)도

BehavioralPatternMapper(행동 패턴 매퍼)(1112), 즉, 행동의 패턴을 결정하는 클래스와 연결되어 있는 것이 바람직하다.

바람직하게, 감정을 생성하는 것은 성공 가능성(Likelihood Of Success: LOS)이 증가 혹은 감소할 때 또는 실패 가능성(Likelihood Of Fail: LOF)이 증가 혹은 감소할 때 감정형 시스템을 통하여 수행된다. 그 성공 가능성이 증가하면, 희망적 감정(hope emotion)이 바람직하게 생성된다. 성공 가능성이 감소하면, 절망하는 감정(despair emotion)이 바람직하게 생성된다. 실패 가능성이 증가하면, 두려운 감정(fear emotion)이 바람직하게 생성되며, 실패 가능성이 감소하면, 즐거운 감정(joy emotion)이 바람직하게 생성된다.

목표의 성공 혹은 실패는 생성된 감정들과 목표 상태에 상당한 영향을 준다. 목표가 실패하면, 절망감이 바람직하게 생성되며, 성공 가능성이 높았다면, 좌절감(frustration)도 바람직하게 생긴다(그 이유는 성공 기대감이 커지기 때문이다).

목표가 성공하면, 즐거움이 바람직하게 생기며, 기대감과 축적된 성공이 높았다면 자존심(pride)이 바람직하게 생긴다.

Emotion(1106)은 메이저와 마이너 타입의 2가지 성질(property)을 갖는 구조이다. 메이저 타입은 경한 감정(minor emotion)이 속하게 되는 상위 레벨 그룹을 기술하는 것으로, 바람직하게는 POSITIVE\_EMOTION(긍정적\_감정)과 NEGATIVE\_EMOTION(부정적\_감정)을 포함한다. 마이너 타입에는 바람직하게, 즐거움(JOY), 희망(HOPE), 흡족함(GLOAT), 자존심(PRIDE), 좋아함(LIKE), 노함(ANGER), 미움(HATE), 두려움(FEAR), 좌절(FRUSTRATION), 고민(DISTRESS), 실망(DISAPPOINTMENT)이 포함된다. 감정의 다른 성질들로는 생성될 때 주어진 그 정도(intensity)와, 쇠퇴 방식(decay policy)(즉, 감정의 변화율)이 있다.

감정 생성 이후의 다음 단계는 지능형 에이전트에 의해 시간의 경과에 따라 생성되었던 감정들을 축적하는 감정적 상태(EmotionalState) 클래스(1104)에 의해 수행된다. 이 클래스는 지능형 에이전트의 현재 감정적 상태를 정의하는 감정 인스턴스들을 수집한 것을 나타낸다. 바람직하게, 현재 감정적 상태는 감정 타입의 계층을 유지함으로써 정의되며, 그 감정 타입들은 집계(aggregation) 및 상관(correlation)에 의해 일반화 된다. 예를 들어, 마이너 감정들은 모아져서 POSITIVE\_EMOTION을 위한 스코어와 NEGATIVE\_EMOTION을 위한 스코어로 되는 것이 바람직하며, 이를 2가지 분류는 GOOD/BAD MOOD(좋은/나쁜 기분)에 상관되어지는 것이 바람직하고, 이것은 지능형 에이전트의 전체적 기분을 기술하게 된다.

EmotionalState 클래스(1104)는 지능형 에이전트 플로팅 애플리케이션에 의해 조회되며; (상술한 모델에서 생성되고, 쇠퇴되고 일반화된 감정들에 의해) 지배적인 행동 패턴이 변화할 때마다, 지능형 에이전트는 바람직하게 그 감정적 상태를 표현하고 그 행동 패턴에 따라 행동한다. 임의로 및 바람직하게, 지능형 에이전트는 텍스트 통신 엔진(이후 더욱 상세히 설명함)과, 3차원 애니메이션과, 얼굴 표정과, 2차원 활동 효과(animated effects) 및 사운드 중 하나 이상을 이용하여 그 감정적 상태를 표현한다.

도 11b는 본 발명에 따른 감정형 시스템에 의해 감정을 생성하기 위한 전형적인 시퀀스 다이어그램을 나타낸 것이다. 도시된 바와 같이, 애플리케이션 관리자(502)(도 5를 참고로 더욱 상세히 설명됨)는 화살표 1로 나타낸 바와 같이 스텝을 FloatingAgentApp(부유 에이전트 애플리케이션)(1006)(도 10을 참고로 더욱 상세히 설명됨)에게 전송한다. 그리고 나서 FloatingAgentApp(1006)는 화살표 1.1에서처럼 목표 클래스(1102)를 조회함으로써 실패 가능성(Likelihood Of Failure: LOF)을 결정한다. 그리고 Goal(1102)은 그 실패 가능성을 결정하며; 새로운 실패 가능성이 이전에 결정된 실패 가능성보다 클 경우에는, 화살표 1.1.1.1에서처럼 감정 클래스(1106)에 요청을 하여 두려움이 바람직하게 생성된다. 이 두려운 감정도 EmotionalState(1104)와의 통신에 의해 감정적 상태에 추가된다.

다음으로, 애플리케이션 관리자(502)는 또 다른 스텝(화살표 2)을 FloatingAgentApp(1006)에게 전송하며, 이것은 화살표 2.1에서처럼 다시 Goal(1102)에게 조회를 하여 성공 가능성(Likelihood of Success: LOS)을 결정한다. 이어서 Goal(1102)은 그 성공 가능성을 결정하고; 만약 그 새로운 성공 가능성이 이전에 결정된 성공 가능성보다 큰 경우에는, 화살표 2.1.1.1에서처럼 감정 클래스(1106)에 요청하여 희망감이 바람직하게 생성된다. 이 희망적 감정도 화살표 2.1.1.2에서처럼 EmotionalState(1104)와의 통신에 의해 감정적 상태에 추가된다.

화살표 3은 FloatingAgentApp(1006)에 또 다른 스텝을 전송하는 애플리케이션 관리자(502)를 나타낸 것으로, 이것은 행위의 실제 성과에 따라 감정의 결정을 요청한다. 만약 그 행위가 실패하였고, 마지막 성공 확률이 0.5 등의 소정 인수보다 크다면(이것은 성공할 것으로 기대되었음을 의미함), FloatingAgentApp(1006)은 화살표 3.1.1.1에서처럼 Goal(1102)로 하여금 Emotion(1106)에 의해 생성된 절망감을 갖도록 한다. 이 절망적 감정도 화살표 3.1.1.2에서처럼 EmotionalState(1104)와의 통신에 의해 감정적 상태에 추가된다. 또한, (바람직하게는 성공 기대감과 무관하게) 행위가 실패하였으면, 화살표 3.1.2에서처럼 Emotion(1106)에 의해 고민이 바람직하게 생성된다. 이 고민 감정도 화살표 3.1.3에서처럼 EmotionalState(1104)와의 통신에 의해 감정적 상태에 추가된다.

다음으로, 애플리케이션 관리자(502)는 또 다른 스텝(화살표 4)을 FloatingAgentApp(1006)에게 전송하며, 이것은 화살표 4.1에서처럼 목표(1102)에게 메시지를 전송함으로써 실제 성공에 기초하여 감정을 생성하게 된다. 이어서, 바람직하게, 목표(1102)는 화살표 4.1.1에서처럼 즐거움이 감정 클래스(1106)에 대한 요청에 의해 바람직하게 생성되도록 한다. 이 즐거운 감정도 화살표 4.1.2에서처럼 EmotionalState(1104)와의 통신에 의해 감정적 상태에 추가된다.

실제 성공도가 예상보다 클 경우에는, Goal(1102)은 바람직하게 자신감이 화살표 4.1.3.1에서처럼 감정 클래스(1106)에 대한 요청에 의해 생성되도록 한다. 이 자신감도 화살표 4.1.3.2에서처럼 EmotionalState(1104)와의 통신에 의해 감정적 상태에 추가된다.

도 11c는 본 발명에 따른 감정적 시스템에 의해 감정을 표현하기 위한 전형적인 시퀀스 다이어그램을 나타낸 것이다. 그러한 표현은 사용자 옵션(preferences)에 의해 좌우되는 것이 바람직하다. 애플리케이션(502)은 스텝(화살표 1)을 FloatingAgentApp(1006)에게 전송함으로써 감정적 표현을 시작시키며, 이것은 화살표 1.1에서처럼 지능형 에이전트의 행동적 패턴에 대해서 bp\_mapper(1108)에게 문의를 한다. 만약 지배적인 행동(dominant behavior)이 변경되었다면, FloatingAgentApp(1006)은 요청을 bp\_display(1110)에 전송하여 행동적 패턴을 설정한다(화살표 1.2.1). bp\_display(1110)는 감정의 실제 표시를 제어한다. 그러면 FloatingAgentApp(1006)은 행위가 행위 관리자(1004)에 대한 메시지에 대기상태(enqueue)가 되도록 요청한다(화살표 1.2.2).

애플리케이션 관리자(502)는 또 다른 스텝(화살표 2)을 FloatingAgentApp(1006)에게 전송하고, 이것은 그 행위가 행위 관리자(1004)에 대한 큐(화살표 2.1)로부터 제거되고 그 행위가 bp\_display(1110)에 의해 행해지도록 요청한다.

#### 제4 절: 사용자와의 통신(Communication with the User)

이 절에서는 본 발명에 따른 사용자와의 통신을 위한 통신 시스템의 바람직한 실시예에 대해서 기술하고 있으며, 이 것에는 텍스트 통신, 오디오 통신 및 그래픽 통신이 포함되며 이것들에 한정되는 것은 아니다. 단지 설명하기 위한 목적으로 한정할 의도 없이 텍스 통신은 그러한 타입의 통신의 일례로서 기술하였다.

도 12는 본 발명에 따른 텍스트 통신을 위한 전형적인 시퀀스 다이어그램을 나타낸 것이다. 텍스트 엔진(1200)은 소정의 이벤트와 관련되고 지능형 에이전트에 의해 전달될 수 있는 텍스트를 생성하는 역할을 한다. 텍스트 엔진(1200)은 바람직하게 미리 결정되며 필러(fillers)를 위한 플레이스 홀더(place holder)들을 포함하는 템플릿(templates)들에 따라 짧은 구(phrase)들 혹은 문장들의 자연 언어 생성을 포함하는 것이 바람직하다. 그 템플릿들과 필러들을 결합함으로써 텍스트 엔진(1200)은 그 템플릿이 속하는 이벤트와 관련되는 많은 수의 구를 생성하게 된다.

이러한 프레임워크는 부가 필러들처럼 부가 템플릿들이 추가될 수도 있기 때문에 많은 새로운 및/또는 변화하는 이벤트들 또는 대상(subjects)들을 위해 임의로 확장될 수 있다.

도 12에 나타낸 바와 같이, FloatingAgentApp(1006)는 우선 바람직하게 특정한 이벤트를 위하여 텍스트를 생성하라는 요청을 전송함으로써 텍스트 엔진(1200)과 통신을 하게 된다(화살표 1). 텍스트 엔진(1200)은 바람직하게 이 이벤트에 적합한 복수의 템플릿들로부터 템플릿을 선택하는 것이 바람직하다(화살표 1.1). 또한 텍스트 엔진(1200)은 바람직하게 이 이벤트에 적합한 복수의 필러들로부터 그 템플릿을 위한 필러를 선택하는 것이 바람직하다(화살표 1.2.1). 이어서 채워진 템플릿은 다시 FloatingAgentApp(1006)으로 전달된다.

다음에 기술하는 것은 지능형 에이전트가 현재 행복한 상태에 있을 경우 몇 가지 전형적인 비한정적인 템플릿과 필러로 기분 변화 이벤트를 위한 텍스트를 생성하는 예이다. 임의로 상기 템플릿들은 다음과 같다:

Happy template 1(행복 템플릿 1): "%noun1 is %happy\_adj2"

Happy template 2(행복 템플릿 2): "%self\_f\_pronoun %happy\_adj1"

임의로, 상기 필러들은 다음과 같다:

```
%noun1 = {"the world(세계)", "everything(모든 것)", "life(생활)", "this day(오늘)", "the spirit(기분)"}
```

```
%happy_adj1 = {"happy(행복한)", "joyful(즐거운)", "glad(기쁜)", "pleased(만족스러운)", "cheerful(유쾌한)", "in high spirits(기분이 꽤 좋은)", "blissful(더없이 행복한)", "exultant(크게 기쁜)", "delighted(아주 기쁜)", "cheery(기분 좋은)", "jovial(명랑한)", "on cloud nine(매우 기쁜)"}
```

```
%happy_adj2 = {"nice(좋은)", "beautiful(멋진)", "great(훌륭한)", "happy(행복한)", "joyful(즐거운)", "good(좋은)", "fun(유쾌한)"}
```

```
%self_f_pronoun = {"I am(나는)", "I'm(난)", "your intelligent agent(너의 지능형 에이전트)", "your agent friend(너의 지능형 친구)"}
```

아래에 기술한 것은 템플릿들과 필러들의 조합으로부터 얻어진 몇 가지 텍스트 통신 구(phrases)의 예이다.

I'm cheerful(나는 유쾌하다)

the spirit is joyful(기쁘다)

I am exultant(나는 매우 기쁘다)

life is beautiful(인생은 아름답다)

life is good(인생은 좋다)

I'm pleased(나는 만족스럽다)

I'm jovial(나는 유쾌하다)

I am joyful(나는 즐겁다)

the world is joyful(세계는 즐겁다)

I'm glad(나는 기쁘다)

the spirit is joyful(기쁘다)

the spirit is happy(행복하다)

the world is nice(세계는 좋다)

I am happy(나는 행복하다)

또 다른 비한정적 예로서 못 받은 전화 템플릿(missed call template)이 다음과 같이 구성될 수 있다.

%user missed a call from %missed %reaction

이 예에서, 사용자의 이름은 %user를 위해 사용되었고; 그 이름 혹은 (예를 들어 전화번호 등의) 다른 식별자가 %missed에 입력되며; %reaction은 선택적인 것으로 예를 들어 실망을 표현하는 것(예를 들어 "나는 슬프다")과 같은 지능형 에이전트의 반응을 위해 사용되는 것이다.

이러한 예들로 나타낸 바와 같이, 텍스트 엔진(1200)은 못 받은 전화 이벤트들에서부터 배터리 부족 이벤트들에 이르는 수많은 이벤트들을 위해 관련된 문장들을 생성할 수 있음으로써, 사용자가 이동 정보 장치와 더욱 풍부하고 더 많이 이해가능하게 상호작용을 할 수 있게 된다.

#### 제5 절: SMS 메시지 및 통화를 위한 적응형 시스템

이 절은 본 발명에 따른 SMS 메시지 및 통화를 적응적으로 처리하기 위한 적응형 시스템의 바람직한 실시예에 대해서 기술하고 있다. 우선 본 발명에 따른 상기 시스템으로 동작하기 위한 몇 가지 바람직한 알고리즘에 대한 일반적인 개요를 설명하기로 하고, 그 다음으로 통화 처리 및 SMS 메시지 처리 클래스 및 시퀀스 다이어그램들에 대해서 설명하기로 한다.

#### *스마트 대안 번호(SAN: Smart Alternative Number)*

SAN 알고리즘은 소정 전화의 시도가 실패한 후에 걸게 될 가능성이 가장 높은 대안 전화번호를 학습하기 위해 고안되었다. 이 알고리즘은 그들 쌍(pair)을 생성하도록 학습하고, 이에 따라 새로운 사용자 행동에 대해 동적으로 적응할 수 있다. 이들 관련 쌍들은 사용자에 의한 전화 시도가 실패한 후에 걸게 될 번호를 제안하기 위해 이용된다.

이 알고리즘은 다음과 같이 임의로 구현될 수 있다. 즉, 가장 자주 이용되는 항목들을 제1 레이어에 삽입하고(임의로, 그 항목의 빈도수가 미리 정해진 기준치보다 크게 되면 그러한 삽입들이 발생된다); 전화번호에 대한 관련 쌍을 바람직하게는 그 리스트 상의 그 쌍의 빈도수에 따라 제안하며; 전화 성공 또는 실패를 결정하고; 번호마다 결정된 쌍들의 히스토리에 대한 윈도를 유지하고, 가장 오래된 것은 임의로 삭제될 수 있도록 하는 것 등에 의해 구현될 수 있다.

이 알고리즘을 위한 지식 베이스는, 대안/후속 전화들의 리스트 및/또는 그 이외의 행해질 행동들을 포함하는, 각 발신 번호를 위한 하나의 숲(forest)으로서 임의로 표현될 수 있다. 각 발신 전화에 대해서, 그 다음 전화는, 첫 번째 전화가 실패하고 두 번째 전화가 소정 기간 이내에 수행되었을 경우, 후속 될 것으로 간주되는 것이 바람직하다. 그 후속 전화번호는 그 첫 번째 전화번호의 그러한 후속 전화번호들의 리스트에 추가된다. 그 리스트가 모두 채워지게 되면, 가장 오래된 전화번호는 바람직하게 잊혀지게 된다.

#### *스마트 전화번호부 관리자(SPBM: Smart Phonebook Manager)*

SPBM 시스템은 이동 정보 장치 사용 통계 및 전화 통계를 이용하여 가능한 연락처들의 관계 및 상호 특성을 학습하는 지능형 전화번호부 시스템의 비한정적 예이다. 이 시스템은 자동화된 연락처 그룹의 생성 및 자동화된 연락처의 추가/삭제 등을 포함하는 (이것에 한정되지는 않음) 여러 가지 새로운 전화번호부 기능들을 제공한다.

예를 들어, 자동화된 그룹 관리 알고리즘은 사용자의 사용방식에 따라 연락처 전화번호를 자동으로 그룹화할 수 있도록 되어 있는 것이 바람직하다.

#### *자동화된 상태 검출(ASD: Automated State Detection)*

ASD 시스템에 의해, 이동 정보 장치는 사용자의 현재 사용 상태(예를 들어 미팅 중, 부재 중)를 결정하여 현재의 상태에 맞도록 사용자 인터페이스, 사운드 및 인공지능 행동 시스템에 대한 변경들을 제안하는 것이 바람직하게 가능하게 된다(예를 들어 착신되는 전화들에 대해서 무음 모드(silent mode)를 동작시키고 및/또는 자동으로 "미팅 중"이라는 SMS 회신을 보냄). 임의로 및 바람직하게, 이 시스템은 하나 이상의 생물학 센서와 통신을 하게 되며, 이 센서는 임의로 및 바람직하게 사용자의 생물학 상태 및/또는 사용자의 이동 등을 감지하게 된다. 이들 부가 센서는, 사용자로부터 특정 입력을 받지 않고 및/또는 사용자들의 현재 상태에 대해 사용자에게 묻지 않고도, 이동 정보 장치를 위해 적응형 시스템이 올바른 상태를 결정할 수 있게 하는 정보를 제공하게 된다. 장치 카메라에 의해 촬영된 화상도 이러한 목적으로 임의로 사용될 수 있다.

또 다른 임의의(optional) 타입의 센서는, 예를 들어 지문 분석 및/또는 다른 타입의 생체인식 정보를 통해, 사용자가 특정 사용자를 식별할 수 있도록 한다. 그러한 정보는 보안상의 이유로 임의로 사용될 수도 있다.

미팅 모드 충고자(advisor) 알고리즘은 사용자가 방해차단모드(do-not-disturb mode)를 관리하는데 도움을 주도록 설계된다. 이 알고리즘은 사용자가 미팅 모드에 있고 방해받고 싶지 않을 확률(사용자가 그 모드를 변경하지 않았고 전화를 받으려고 하는 확률의 반대)을 나타내는 규칙베이스를 가지고 있다. 이 알고리즘의 목적은 그러한 변환(transitions)의 관리를 돋고자 하는 것이다.

바람직하게, 이 알고리즘은 전화기 세계 상태를 결정하고, 또한 (예를 들어, 사용자가 현재 전화벨을 정지하고 그 전화에 응답을 하지 않는 등의) 미팅모드가 제안되어야 한다고 규칙베이스가 지시하는 시기를 결정함으로써 상술한 바와 같은 인공지능 상태 처리기를 통해 동작하게 된다. 또한, 바람직하게 그 StateHandlers(상태핸들러)는 또한 미팅모드가 취소되어야 한다고 지시할 수 있는 그 반대 타입의 이벤트들에 주의를 기울이게 된다.

도 13a 및 도 13b는 각각 본 발명에 따른 통화 처리를 위한 전형적인 클래스 다이어그램과 전형적인 시퀀스 다이어그램을 나타낸 것이다.

도 13a에 나타낸 바와 같이, 통화 처리 클래스 다이어그램(1300)은 SuggestCall(전화 제안) 클래스(1304)에 의한 SuggestCall 행위들의 생성을 담당하는 CallStateHandler(전화상태핸들러)(1302)를 특징으로 가지고 있다. 바람직하게 CallStateHandler(1302)는 CallStartedEvent(통화 시작 이벤트)(1306)와, CallEndedEvent(통화 종료 이벤트)(1308)와, CallFailedEvent(통화 실패 이벤트)(1310)와 같은 통화 이벤트에 주위를 기울이는 규칙 기반 알고리즘이며, 그리고 이들 이벤트는 각각 CallEvent(1312) 클래스와 통신을 한다. 또한 바람직하게 CallStateHandler(1302)는 다음과 같은 2가지 주요 기능을 담당하는 규칙 베이스를 보유한다. 2개의 주요 기능 중 하나는 기계 학습(Machine Learning)으로서 전화 연관사항 지식베이스(call associations knowledge base)를 보유하는 것이며, 다른 하나는 사용자에게 통화할 전화번호를 제안할지 여부에 대한 인공지능 확률 기반 추정(AI probability based inference)이다. 그러한 제안들은 SuggestFollowingCall(후속 전화 제안)(1314) 혹은 SuggestAlternativeCall(대안 전화 제안)(1316)을 의해 처리되는 것이 바람직하다.

통화 이벤트 객체들은 다시 이동 정보 장치의 운영체제의 후크 기능에 의해 상술한 바와 같은 이벤트 모델을 이용하여 생성된다. 그 통화 데이터에는, 만약 가능하다면, (전화번호, 연락처 이름, 시작 시간, 기간(duration) 등의) 생성된 이벤트에 대한 정보로 채워지는 것이 바람직하다.

전화 제안 클래스들(1304, 1314 및 1316)은 적응형 행위 모델에서 기술된 베이스 행위 인터페이스를 구현한다. 이를 클래스는 소정의 통화가 종료한 후에 후속 전화를 걸기 위한 전화번호를 사용자에게 제안하거나 소정의 통화가 실패한 후에 대안 통화를 하기 위한 전화번호를 사용자에게 제안하는 역할을 한다.

CallStateHandler(1302)는 통화 이벤트들에게 주위를 기울이며 그 이벤트를 그 규칙 베이스에 따라 분류한다(규칙 베이스 방식을 이용한 행위 선택). 전형적인 임의의 전화 제안 규칙 베이스는 다음과 같이 주어진다.

1. 통화가 시작되었고 마크된 이전 통화(prior call)가 없으면 -> 통화를 시작된 것으로 마크함
2. 그 통화가 종료되었고 통화가 시작된 것으로 마크되었으면 -> 첫 번째 통화로 마크함
3. 통화가 시작되었고, 그 이전 통화가 첫 번째 통화로 마크되었으며, 통화들 사이의 시간이 후속 통화 기준치(following call threshold)보다 작으면 -> 후속 전화로서 마크함
4. 통화를 후속하는 것으로 마크하였으면 -> 지식베이스를 갱신함
5. 통화를 후속하는 것으로 마크하였으면 -> 통화를 첫 번째 통화로 마크함(이에 따라 다음 통화와의 비교를 위해 현재의 첫 번째 통화가 리셋됨)
6. 통화가 실패하였고 마크된 이전 통화가 시작된 통화로서 마크되었으면 -> 통화를 처음으로 실패한 것으로 마크함
7. 통화가 시작되었고 이전 통화가 처음으로 실패한 것으로 마크되었고 통화들 사이의 시간이 실패 전화 기준치(failed call threshold)보다 작으면 -> 대안 통화로 마크함
8. 대안 통화로서 마크하였으면 -> 지식 베이스를 갱신함

9. 통화가 종료하였고 종료 통화 이후의 시간(time since end call)이 제안 후속 통화 기준치(suggest following call threshold)보다 작고 후속 통화를 추론하였으면 -> 제안된 후속 통화 행위(예를 들어 그 통화 이후에 취해지는 행위)를 생성함

10. 통화가 실패하고 실패한 통화 이후의 시간이 제안된 대안 통화 기준치보다 작고 후속 통화를 추론하였으면 -> 제안된 대안 통화 행위를 생성함

11. 마지막 마크로부터의 시간(time from last mark)이 기준치보다 크면 -> 모든 통화의 마크를 취소(unmark)함

통화 제안 지식 베이스는 히스토리 윈도(history window)로서 임의로 및 바람직하게 설계되며, 이때 연관사항들(associations)이 그 서브젝트의 히스토리에 발생사항들(occurrences)로서 추가된다. 이 경우, 소정의 연락처에 대한 대안 혹은 후속 통화를 위한 관련된 서브젝트는 그 연락처 자체이며, 모든 연관된 통화들은 그 대안 통화 혹은 후속 통화 히스토리 윈도에 발생순서(occurrence order)에 의해 바람직하게 배치된다.

예를 들어 연락처 전화번호가 054-545191인 경우, 히스토리 윈도 내의 통화 연관사항들은 다음과 같이 임의로 제공될 수 있다.

054-545191 ->

052-	052-	051-	052-	051-	052-	051-	051-	052-	052-
552211	552212	546213	552211	546213	552211	897555	897555	552211	552211

그 히스토리 윈도 크기는 상기 알고리즘이 관리(혹은 기억)하게 되어 있는 연관사항들(발생사항들)의 개수로서 상기 알고리즘에 의해 정의되는 것이 바람직하다. 히스토리 윈도가 모두 채워지면, 새로운 발생사항이 그 맨 앞에 바람직하게 추가되어 마지막 발생사항은 제거되게 되며, 따라서 윈도는 그 정의된 크기를 넘지 않게 된다. 그 지식 베이스는 더 최근의 연관사항들을 우선으로 하여 오래된 연관사항들은 제거되게(잊혀지게) 되므로 사용자 패턴에 있어서의 변화에 적응할 수 있도록 되어 있다.

지식 베이스는 대안 혹은 후속 통화를 제안할 만큼 충분한 것은 아니며, 그 이유는 좋은 제안이 지식 베이스로부터 추론되어야 할 필요가 있기 때문이다. 바람직하게 그 추론 알고리즘은 지식 베이스에 따라 가능성이 제일 큰 연관사항 타겟을 결정하기 위해 단순 확률 기반 추론(simple probability based inference)이다. 다음과 같은 파라미터를 고려하면:

$C_0$  - 상수

$H_0(C_i)$  - 연락처( $C_i$ )가  $C_0$  히스토리 윈도에서 발견될 수 있는 횟수

$H_{size}$  - 히스토리 윈도 크기

바람직하게 상기 방법의 제안에 따르면 연락처  $C_i$ 는 다음과 같이 된다.

$$P(C_i) = \text{Max}(H_0(C_i)/H_{size})$$

상술한 예에서는  $C_0 = 054-545191$ 일 경우에:

$P(052-552211)$ 가 최대이고 0.6이다.

추론 과정은 %50보다 큰 확률을 추론할 수 있을 경우에만, 바람직하게는 히스토리 윈도가 또한 모두 채워지면, 성공적인 것으로 간주된다.

도 13b는 통화 처리를 위한 전형적인 시퀀스 다이어그램을 나타낸 것이다. 도시된 바와 같이, EventDispatcher(이벤트 발송자)(604)(도 6에 더욱 자세히 기술됨)는 통화 이벤트에 대한 통지를 CallStateHandler(통화 상태 핸들러)(1302)에게 전송하며(화살표 1), 이어서 이것은 그 규칙 베이스를 평가한다. 연관사항을 HistoryWindow(히스토리 윈도)(1318)에 추가하라는 요청이 이루어진다(화살표 1.1.1).

통화 종료된 혹은 실패된 이벤트를 위해서, EventDispatcher(604)는 통지를 CallStateHandler(1302)에게 전송하며(화살표 2), 이어서 이것은 규칙 베이스를 평가한다(화살표 2.1). 연관사항을 HistoryWindow(1318)에 추가하라는 요청이 이루어진다(화살표 2.1.1). HistoryWindow(1318)로부터 가망성 있는 연관사항(probable association)을 수신하라는 요청이 이루어진다(화살표 2.2). 이 가망성 있는 연관사항은 예를 들어 통화할 전화번호를 나타내며, CallStateHandler(1302)에서 SuggestCall(1304)로 전송된다(화살표 2.3). 이 행위는 행위 관리자(1008)에 의해 대기상태가 된다(화살표 2.4).

또 다른 임의의 그러나 바람직한 알고리즘은 사용자가 못 받은 전화(missed calls) 및 전화 대기 기능(call waiting function)을 관리하는데 도움을 준다. 이 알고리즘은 못 받은 중요한 전화들(전화를 기다리는 도중에 발생가능)을 식별하여 지능형 회신(intelligent callback)을 제안하는 것에 목표를 두고 있다. 이 회신은 사용자에 의해 (또는 지식베이스에 의해) 중요한 것으로 식별된 번호들에게만 제안된다.

그러한 지식 베이스는 2가지 가능한 (보완적인) 옵션들에 기초하고 있다. 제1의 옵션은 명시적(explicit)인 것으로, 이 경우 사용자는 확장된 주소록 영역 내의 다른 정보와, 수행된(전화를 건) 후의 그 전화의 중요성을 지시하게 된다. 제2의 옵션은 비명시적(implicit)인 것으로 그 전화의 빈도수와 다른 변수들에 의해 주어지게 된다.

그러한 알고리즘은 회신 번호가 중요한 것이고 사용자가 일정 기간동안 그 전화를 하지 않고, 및/또는 그 대상 번호가 전화를 걸어오지 않은 경우 회신(call back)을 제안할 수 있다.

#### **콘텐츠 방식 SMS 수신자 추측 (CSAI: Content-based SMS Addressee Inference)**

임의로 및 바람직하게, CSAI 알고리즘은 메시지의 콘텐츠에 의해 그 메시지 수신자를 예측하도록 고안되었다. 이 알고리즘은 바람직하게 메시지의 특정 단어의 패턴들을 식별하여 그것들을 어느 기준의 주소록 연락처와 관련시키도록 학습한다. 이 연락처는 메시지작성 완료시에 메시지 목적지로서 제시된다.

이 알고리즘은 하나 이상의 규칙에 따라 임의로 수행될 수 있으며, 그리고 이 규칙들은 규칙 번역부(rule interpreter)에 의해 번역되게 된다. 적응형 시스템은 (예를 들어 학습모듈을 통해) 예를 들어 1000 단어의 테이블을 학습하는 것이 바람직할 것이다. 발신되는 SMS 메시지내에 보여지는 새로운 단어는 각각 그 리스트에 추가된다. 단어 각각에 대해서, SMS 연락처들(즉, 적어도 하나의 SMS가 전송되었던 연락처들) 각각을 위한 입력항(entry)이 존재한다. 각 단어/연락처 입력항에는 그 연락처로 전송된 SMS 메시지들에서 그 단어가 보여지는 회수가 포함되며, 바람직하게는 각 연락처로 전송된 SMS 메시지의 개수도 함께 포함된다.

이러한 추측 메커니즘이 동작하기 위해서는, 바람직하게 현재 SMS 메시지에 있는 각 단어(W)와 각 연락처(C)에 대해서, 그 테이블에 주어진  $P(W|C)$ 와 그 테이블로부터 또한 연산된  $P(C|W)$ 에 기초하여 확률  $P(C|W)$ 이 계산된다. 그리고, 현재 SMS 메시지내의 단어들의 개수에 의해 정규화된 용어의 개수가 추가된다.

상기 SMS 처리 방법은 SMS 메시지 콘텐츠를 분석하고 "전송할"수신자 어드레스를 추론하는 것을 목적으로 한다. 임의로 및 바람직하게, 상기 알고리즘은 다음과 같은 발견적 방법(heuristics)을 이용하는데, 이를 위해 메시지를 특정 수신자에게 보낼 경우 다시 나타나게 되는 특정 지시 단어들(specific indicating words)을 이용한다.

발신되는 SMS 메시지에 나타나는 새로운 단어가 각각 그 리스트에 추가된다. 단어 각각에 대해서, SMS 연락처들(즉, 적어도 하나의 SMS가 전송되었던 연락처들) 각각을 위한 입력항이 존재한다. 각 단어/연락처 입력항에는 그 연락처로 전송된 SMS 메시지들에서 그 단어가 보여지는 회수가 포함된다. 또한 바람직하게는, 각 연락처에 전송된 SMS의 개수가 저장된다. 새롭게 보내진 SMS 메시지들을 파싱(parsing)한 후에 그 단어 테이블을 갱신함으로써 학습이 바람직하게 발생하게 된다.

인공지능 추론 방법은 상술한 바와 같이 단순 확률에 의해 바람직하게 이루어지게 된다.

도 14a 및 도 14b는 각각 본 발명에 따른 SMS 메시지 처리 클래스 및 시퀀스 다이어그램들의 전형적인 비한정적 예들을 나타내고 있다.

도 14a는 본 발명에 따른 전형적인 SMS 메시지 처리 클래스 다이어그램(1400)을 나타내고 있다. SMSstateHandler(SMS 상태 핸들러)(1402) 클래스 및 SuggestSMSToSnd(SMS 전송 제안)(1404) 클래스가 도시되어 있다. SMSstateHandler(1402)는 SMS를 보내는 상태에 대한 정보를 수신하는 역할을 하며; 그리고 SuggestSMSToSnd(1404)는 SMS를 전송하게 되는 수신측 어드레스(전화번호)를 제안하기 위해 접속된다.

도 14b는 그러한 제안을 행하기 위한 전형적인 시퀀스 다이어그램을 나타내고 있다. EventDispatcher(604)(더 자세한 설명은 도 6을 참고)는 SMS 이벤트에 대해서 SMSstateHandler(1402)에게 통지를 보낸다(화살표 1). SMSstateHandler(1402)는 지식 베이스를 파싱하는 것에 의해 시작되며(화살표 1.1.1); 그리고 나서 연락처들에 대한 정보를 위해서 요청이 SMSdata(1406)에게 전송된다. SMS는 화살표 1.1.1.2에서처럼 바람직하게 토큰화되고(예를 들어 파싱되고), 제안된 연락처 어드레스가 SMSdata(1406)에게 요청된다(화살표 1.1.1.3).

제안된 연락처 어드레스가 얻어지면, SMSstateHandler(1402)는 행위를 바람직하게 생성하고, 이것은 SuggestSMSToSnd(1404)에게 전송되며(화살표 1.1.2.1.1), 이어서 이 행위에 대한 목표가 설정되고(화살표 1.1.2.1.2), 행위 관리자(1008)(더 자세한 설명은 도 10을 참고)에게 전송함으로써 그 행위가 대기상태가 된다(화살표 1.1.2.1.3).

일단 SMS가 전송되면 통지가 EventDispatcher(604)에서 SMSstateHandler(1402)에게 전송되며(화살표 2), SMSstateHandler(1402)은 그 상태를 처리하며(화살표 2.1), 이 처리에는 지식 베이스를 갱신하고(화살표 2.1.1), 새로운 SMS 데이터를 그 안에 삽입하는 것(화살표 2.1.1.1)이 바람직하게 포함된다(이 처리는 SMSdata(1406)과 통신을 하면서 행해짐).

#### **제6 절: 메뉴용 적응형 시스템**

이 절에서는 본 발명에 따른 메뉴의 적응적 처리를 위한 적응형 시스템의 바람직한 실시예에 대해서 기술하기로 한다. 우선 메뉴를 구성하고, 배열하면 재배열하기 위한 알고리즘의 일반 개요에 대해서 설명하고, 그 다음으로 전형적인 메뉴 처리 클래스 다이어그램(도 15)에 대해서 설명하기로 한다.

그러한 적응형 메뉴 시스템은 자동적 추론을 이용하는 것에 의해 이동 정보 장치의 운영체제가 제공된 휴먼 유저 인터페이스 혹은 메뉴 시스템을 고객맞춤화하는 능력을 기초로 하고 있다. 그래서 유저 인터페이스를 구비한 모든 운영체제는 메뉴, 원도 혹은 그에 상당하는 사용자 인터페이스 시스템을 가지고 있다. 운영체제들 중 많은 수가 특정 사용자를 위한 메뉴시스템 혹은 원도시스템을 수동으로 혹은 관리적(administrative)으로 고객맞춤화하는 옵션을 가지고 있다. 상술한 시스템은 사용자 인터페이스를 자동으로 고객맞춤화할 수 있는 가능성을 제공한다. 자동적 행위들이 상술한 시스템에 의해 생성되는 것이다(사용자의 승인으로 혹은 자동적으로 가능함). 상기 시스템은 메뉴 시스템 프레임워크를 이용하여, 올바른 고객맞춤화 행위를 추론하고 또한 운영체제가 제공된 고객맞춤화 옵션들을 자동적으로 이용하는 능력을 제공하는데 필요한 지식베이스와 추상화(abstractions)를 제공한다.

### **지능형 메뉴 어셈블러(IMA: Intelligent Menu assembler)**

IMA 알고리즘은 특정한 사용자 옵션(preferences) 및 이동 정보 장치의 사용방식(usage)에 기초하여 사용자 인터페이스(UI) 메뉴를 동적으로 생성하도록 고안되었다. 이 알고리즘은 바람직하게 전화사용 특성들을 식별하고 그러한 특성들에 기초하여 특별한 개인용 메뉴를 형성한다.

그리고, 이러한 알고리즘은 그러한 메뉴를 구성하기 위한 두 개의 다른 알고리즘을 특징적 요소로서 임의로 가질 수 있다. 자동 메뉴 단축키 메뉴 알고리즘은 선호하고 및 가장 자주 이용되는 애플리케이션들 및 서브 애플리케이션에 대한 자동 단축키 혹은 단축수단(shortcut)들을 생성하는 것에 목적이 있다. 이 알고리즘은 그들의 개인용 단축키 메뉴를 설정하기 위해, 사용자들 대부분이 사용하지 않았던 수동 방식을 향상시키는데 중점을 두고 있다. 지식베이스 및 학습을 위해, PhoneWorldMapper(전화 세계 매퍼)는 애플리케이션들 및 서브 애플리케이션들의 실행들을 축적하고, 그 지식을 어떠한 애플리케이션/서브 애플리케이션에게 개인용 메뉴 옵션의 메뉴 단축키를 부여하여 사용자를 위해 설정해야하는지를 추측하기 위해 사용한다.

이러한 단축키 추론은 다음과 같은 효용 함수(utility function)에 기초하여 행해진다. 즉, 단축키에 의해 절약되는 필요한 클릭 양(일반 메뉴의 클릭수(빼기) 단축키상의 클릭수)에 의해 가중된(weighted) 사용되는 애플리케이션의 빈도수에 기초하여 행해진다. 최고 효용(highest utility)의 애플리케이션/서브애플리케이션/스크린은 사용자에게 그 제안과 단축키 구성을 제공하게 된다.

또 다른 그러한 메뉴 알고리즘은 그 자동 메뉴 재구성 알고리즘을 임의로 포함할 수 있다. 이 알고리즘이 타깃으로 하고 있는 것은 메뉴 시스템에 있어서 개별 맞춤화의 부족함에 있다. 많은 사용자들은 전화기의 사용자 인터페이스를 각기 다른 방식으로 사용하지만 동일한 메뉴 시스템 및 인터페이스를 가지고 있다. 이 알고리즘은 사용자의 특정한 사용방식을 학습하여 그에 따라 메뉴시스템을 재구성함으로써, 더욱 바람직하게 완전한 적응형 메뉴시스템을 제공하게 된다.

이러한 지식베이스를 위해, PhoneWorldMapper(전화기 세계 매퍼)는 특정 타겟에 대한 클릭 수를 절약할 뿐만 아니라, 애플리케이션들 및 서브 애플리케이션들의 실행들을 축적한다. 이 전화기 세계 매퍼는 아이템들을 동일한 메뉴에 추가할 때 사용되는 계층적 시야(hierarchical view)를 부여할 것이다. 메뉴 내에서는, 그 아이템들이 그들의 효용성에 의해 구성되어진다.

바람직하게, 상기 메뉴 시스템은 (위에서 정의된 변수들에 의해) 사용자에게 최적인지 여부가 주기적으로 평가되며, 임의로, 추론된 최상의 옵션에 따라 그 평가에 후속하여 재구성되어진다.

도 15는 적응형 메뉴 시스템 클래스 다이어그램(1500)을 나타내고 있다. 이 클래스 다이어그램은 PhoneWorldMapper(1502) 클래스와 PhoneWorldNode(전화기 세계 노드)(1504) 클래스를 통해 필요한 추상화를 제공한다. PhoneWorldMapper(1502)는 메뉴 및 사용자 인터페이스 시스템을 매핑하는 역할을 한다. 이 매핑은 PhoneWorldNode(1504)와 함께 이루어진다. PhoneWorldNode(1504)는 그래프 구조의 메뉴, 서브메뉴 혹은 메뉴 아이템을 나타낸다.

바람직하게 PhoneWorldMapper(1502)는 PhoneWorldNode(1504) 객체들의 그래프를 포함하며; 그 에지들은 그 노드들 사이의 메뉴 전이부(menu transitions)에 해당하고 버텍스들은 매핑된 메뉴들과 아이템들이다. 사용자가 메뉴 시스템 내에서 네비게이트할 때마다, PhoneWorldMapper(1502)는 PhoneWorldNode(1504) 객체들의 객체 그래프를 통해 따라다니며, 그것을 사용자의 올바른 위치로 포인트한다. 사용자가 소정의 아이템을 활성화시킬 때마다, 현재 노드는 그 행위 및 카운트 활성화를 바람직하게 기록한다. PhoneWorldMapper(1502)는 본 발명이 각 메뉴 아이템 사이의 (클릭에 있어서의) 거리를 산출하는 능력과, 루트로부터의 그 거리를 산출하는 능력을 갖도록 하며, 이것은 상기 메뉴 시스템의 그래픽 표현 때문에 가능하다. 따라서, PhoneWorldMapper(1502)는 메뉴에 있어서의 아이템들 사이의 거리와, 메뉴 활성화와, 메뉴 네비게이션과 메뉴 구조에 대한 추상화(abstraction)를 제공한다.

또한 바람직하게 클래스 다이어그램(1500)은 메뉴를 통해 단축기를 제안하기 위한 SuggestShortcut(단축기 제안)(1508)와 메뉴 이벤트들을 처리하기 위한 MenuEvent 클래스(1506)를 포함한다. 바람직하게, PhoneWorldMapper(1502)는 메뉴와 관련하여 사용자의 개인적 사용 패턴을 기술하기 위한 MyMenuData(내 메뉴 데이터) 클래스(1510)와 통신을 하고 또한 상술한 그래프를 위한 메뉴 노드들을 제공하기 위한 PhoneWorldMenuNode(전화기 세계 메뉴 노드)(1512)와 통신을 한다. PhoneWorldLeafNode(전화기 세계 리프 노드)(1514)는 또한 상술한 그래프를 지원하기 위하여 PhoneWorldNode(1504)와 통신을 한다.

상술한 바와 같은 시스템은 적응형 사용자 인터페이스 알고리즘들의 레벨들을 제공한다. 제1 고객맞춤화 레벨은 메뉴 아이템 활성화 단축키 제안 알고리즘을 바람직하게 특징으로 갖는다. 이 알고리즘은

PhoneWorldMapper(1502)를 이용하여 아이템들의 활성화들을 감시한다. 이 알고리즘은 메뉴 아이템의 활성화의 평균 횟수를 감시한다. 소정 아이템에 대한 다수의 활성화들이 기준치를 넘고(임의로 평균을 넘고), 단축키 활성화의 거리가 그 자신의 아이템 활성화에 필요한 것보다 짧으면, 단축키가 바람직하게 제안된다. 사용자는 이러한 자동화된 단축키로부터 이득을 얻게 되는데, 그 이유는 그것에 의해 사용자가 원하는 기능을 동작시키기 위해 수행하는 동작 횟수가 감소하기 때문이다. 이러한 흐름을 기초로, 행위 생성은 그 지식 베이스로서 PhoneWorldMapper(1502)를 사용하는 규칙 기반 방식(rule based strategy)이다. 이러한 알고리즘은 사용자 특정 단축키들을 자동으로 고객맞춤화하게 된다.

제2 고객맞춤화 레벨은 메뉴 아이템 재배열을 바람직하게 포함한다. 그 알고리즘은 PhoneWorldMapper(1502)를 사용하여 아이템들의 활성화를 감시하고 그 활성화 횟수에 따라 특정 메뉴 안의 아이템들을 재배열하여, 가장 자주 사용되는 아이템들을 처음에 보이게 한다. 이 알고리즘은 사용자의 특정한 사용방식에 적응함으로써 메뉴 아이템들의 순서를 고객맞춤화한다. 이 알고리즘은 메뉴 아이템들을 재배열하기 위해 앞의 알고리즘과 동일한 활성화의 지식 베이스를 바람직하게 사용한다.

제3 고객맞춤화 레벨은 메뉴 구성(composition)을 바람직하게 포함한다. 그 알고리즘은 메뉴들과 아이템들의 사용을 감시하여 가장 많이 사용된 아이템들을 선택한다. 이를 아이템들을 위해서, 그 알고리즘은 PhoneWorldMapper(1502) 그래프 내에서 첫 번째 공통 노드를 선택한다. 이 노드는 메뉴가 되며 가장 많이 사용된 아이템들은 그 노드의 메뉴 아이템들이 된다. 이 메뉴는 메뉴 시스템에서 첫 번째로 배치되는 것이 바람직하다. 이 것에 의해, 또한 PhoneWorldMapper 그래프가 그 메뉴 시스템에서의 변화를 나타내는 새로운 그래프로 변경되게 된다. 바람직하게 이 알고리즘은 아이템 활성화의 내림차순으로 메뉴들을 반복하여 구성한다.

## 제7 절: 게임용 적응형 시스템

이 절에서는 본 발명에 따른 게임용 적응형 시스템의 바람직한 실시예에 대해서 기술하기로 한다. 본 발명에 따른 전형적인 게임 클래스 다이어그램이 도 16에 도시되어 있다.

임의로 및 바람직하게, 지능형 에이전트의 목표들 중 몇몇은 사용자를 즐겁게 하기 위한 것이다. 또한 임의로 상기 지능형 에이전트는, 예를 들어 정보교환(communication)을 하기 위해, 개인적 목표들(personal goals)을 가지고 있다.

이들 목표의 상태를 관리하기 위해서, 바람직하게 상기 시스템은 도 16의 게임 클래스 다이어그램(1600) 안에 2개의 클래스(UserBoredStateHandler(사용자 지루한 상태 핸들러)(1602) 및 CreatureStateHandler(생물 상태 핸들러)(1604))를 바람직하게 가지고 있다. 바람직하게 이 2개의 클래스는 모두 규칙 기반 방식에 따라 행위들을 생성한다. 이 클래스들에 의해 유지되는 규칙은 그들에 의해 표현되는 목표들에 관한 것이다. 상기 2개의 클래스는 모두 상태 변화와 규칙들의 평가를 위한 입력 방법으로서 이벤트 모델을 이용한다(즉 2개가 모두 이벤트 핸들러임).

Idle(유휴) 행위로서(즉 달리 관여하지 않을 경우), 지능형 에이전트는 MoveAction(이동 행위)(도시하지 않음)을 바람직하게 선택하며, 이것은 또한 예를 들어 애니메이션 및 소리 등과 관련하여 사용자 성향(preferences)에 적용될 수도 있다.

바람직하게 MoveAction은 우선 MOVE(이동) 상태와 REST(휴식) 상태 사이에서 선택하게 된다. 각각의 상태에 있어서, MoveAction은 확률(probability)을 기초로 하여 그 상태를 위한 적절한 애니메이션을 선택한다. 그 확률은 각 옵션에 대해서 50%로 초기화된다.

사용자 입력은 현재 선택된 쌍(상태 및 애니메이션)에 대한 확률에 영향을 준다. 사용자가 그 상태의 확률에 나쁜 입력을 부여하면 그 현재 애니메이션 및 그 상태의 확률이 감소하며, 좋은 입력을 부여하면, 그 확률은 감소하게 된다. 바람직하게 그 확률은 최소 및 최대 값에 대한 소정의 기준치를 가지고 있어서 소정의 상태 혹은 애니메이션이 결코 선택될 수 없게 될 가능성을 방지하게 된다.

또한 CommAction(1606)이 도시되어 있다. 이 행위는 통신(정보교환)하기 위한 목표에 의해 유도되며, 지능형 에이전트 통신 상태와 사용자의 표현력 및 통신 성향에 따라, CreatureStateHandler(1604)에 의해 임의로 생성된다. 예를 들어, 지능형 에이전트가 일정 시간 동안 사용자와 통신을 하지 않았고, (상태 핸들러 규칙 베이스에 따라) 현재가 사용자와의 통신을 시도하기에 좋은 시간이면, 통신 행위가 바람직하게 생성된다. 이 행위는 진동 혹은 소리를 일으킬 수 있으며, 또한 가능하다면 텍스트 통신이 임의로 사용될 수도 있다.

임의로 및 바람직하게 행동 표시 행위는 감정형 모델에 의해 동작하게 되며; 감정적 상태가 변화할 때마다 지능형 에이전트는 (텍스트, 소리, 2차원 및 3차원 애니메이션 등을 임의로 사용하여) 새로운 감정적 상태를 바람직하게 표현한다.

GameAction(게임 행위)(1608)은 부유(floating) 애플리케이션 공간상에서 게임을 시작시킨다. 이 행위는 가상 인공지능 세계 애플리케이션으로부터 하나 이상의 객체를 임의로 선택한다. 지능형 에이전트는 객체를 탐색하고 그것에 따라 행동하게 된다. 예를 들어 공이 선택될 수 있고, 공이 선택되면 지능형 에이전트는 그 공을 이동시키거나 찰 수 있고, 사용자는 그 공을 새로운 장소로 이동시키는 등을 할 수 있다. 그 객체들 중 어떤 것들은 포장된 사용자 인터페이스 객체들(wrapped user interface objects)이 될 수 있다(인공지능 세계 애플리케이션에 기술되어 있음). 바람직하게, 이 게임 행위는 그것이 단지 지능형 에이전트이며 사용자의 보강 없이 가능한 행동의 선택을 결정한다는 특징을 갖는다.

HideAndSeek(숨바꼭질) 행위(1610)는 메뉴시스템과, 다른 호스트 스크린들에서 사용자의 위치를 추적하기 위해 PhoneWorldMapper의 능력을 이용한다. 바람직하게, 지능형 에이전트는 메뉴 트리에서 위치를 선택하고 숨고, 그

후 사용자는 지능형 에이전트를 찾을 때까지 혹은 탐색시간이 끝날 때까지 메뉴 시스템을 네비게이트한다. 사용자가 지능형 에이전트를 (찾지 못하거나) 찾은 후에, 바람직하게, 메뉴 시스템에서의 현재 위치에 대한 무엇 및/또는 현재 스크린에 대해서 도움이 되는 무엇을 사용자에게 알려주는 메시지가 포스트된다. 이러한 방식으로, 사용자는 그 호스트 플랫폼에서 이용 가능한 특징들과 다른 옵션들에 대해서 학습할 수 있게 된다. 바람직하게, 도움이 되는 툴팁(too-tips)은 그 클래스의 객체 인스턴스에 의해 기술된 특정 노드에 관련한 툴팁을 포함하는 PhoneWorldNode를 통하여 지능형 에이전트에 의해 이용 가능하다.

SuggestTmTrivia(1612)는, 바람직하게는 사용자가 관심이 있다고 표현했었던 토픽에 대해서, 사용자에게 트리비아 게임을 임의로 제공할 수 있다.

#### 제8 절: 교육 시스템 (Teaching System)

이 절에서는 본 발명에 다른 교육 시스템의 바람직한 실시예에 대해서 기술하기로 하며, 이것에는 장치 자체의 동작과 직접 관련되지 않은 서브젝트에 대해서 사용자에게 가르치기 위한 본 발명의 바람직한 실시예도 포함되며 이것에 한정되는 것은 아니다. 그러한 교육 기계에 대한 일반 개요에 대해서 설명하기로 하며, 이어서 도 17a(전형적인 교육 기계 클래스 디어그램) 및 도 17b(전형적인 교육 기계 시퀀스 디어그램)에 따른 교육 기계의 임의의 그러나 바람직한 구현에 대해서 기술하기로 한다.

상술한 애플리케이션 레이어는 본 발명의 프레임워크 내의 서로 다른 교육 애플리케이션들을 생성하기 위해 그 교육 기계의 기반구조(infrastructure)를 바람직하게 이용한다.

상기 교육기계는 교육 및 학습의 콘텐츠, 교육 로직, 저장, 개선, (만일 그것이 있다면) 지능형 에이전트와의 상호작용, 레슨(lesson) 구성, (가정 단어들이 소리 내어 읽히거나 이해되어질 경우) 발음 등과 같은, 교육 및 학습의 여러 요소(aspect)들을 위한 지원을 처리 및/또는 제공할 수 있는 것이 바람직하다. 후자의 경우(latter issue)는 언어 정의(이름, 문자 세트, 모음 등); 규칙(문법, 구문론) 및 단어모음(vocabulary) 등의 데이터가 각 언어를 위해 저장될 필요가 있으므로, 언어를 교육시키기 위해 특히 중요한 것이다. 바람직하게, 하나의 규칙은 예를 들어 교육을 받을 수 있고 쉽게 유효성이 확인되는 하나의 단순 언어 요소에 해당한다. 바람직하게, 단어모음은 여러 세트의 단어들로서 정의되며, 여기에서 각 단어 세트는 하나의 수준을 가지며, (업무 단어들, 여행 단어들, 단순화 단어들 등과 같은) 다른 판단기준에 따라 임의로 분류될 수도 있다. 다른 중요 요소(aspect)들에는 문맥(context)이 포함되어, 각 단어(w)에는 적어도 3개의 문맥들과 관계들(relations)이 있어야 하고, 따라서 단어집 내에 있는 단어들인 w1, w2 각각에 대해서, 최대 세트의 관계들이 있어야 한다. 바람직하게, 하나의 관계는 한 세트의 4개의 단어(w1:w2 like w3:w4, 즉, w3:w4와 유사한 w1:w2)로서 정의된다.

고수준 교육기계 구조는 TMLanguage라고 불리는 클래스를 포함하는 것이 바람직하며, 이것은 현재 TM 언어를 위한 추상화(abstraction)를 제공하고 전체 TM 기반구조를 위한 확장 기능들을 허용하게 된다. 또한 바람직하게, TMLesson으로 정의되는 클래스가 있으며, 이것은 예를 들어 한 세트내의 단어들, 규칙들, 퀴즈들 또는 연습문제들 등에 따라, 개별 레슨들을 조직하기 위한 것이다.

레슨 주기는 일주일로 임의로 정의된다. 하나의 레슨은 이 레슨을 위한 현재 단어모음으로서의 하나의 단어 세트와; 이 레슨에 의해 교육되는 하나 이상의 규칙들을 포함할 수 있는 하나의 규칙 세트와; 사용자가 그 자료(material)를 연습할 수 있도록 하는 연습과; 임의로 하나의 퀴즈로 구성되어 있다.

도 17a는 포괄적(generic)이고 적응적인 교육 애플리케이션을 위한 확장 가능한 프레임워크를 제공하기 위해 설계된, 교육 기계 기반구조를 위한 전형적인 교육 기계 클래스 디어그램(1700)을 나타내고 있다. 이 애플리케이션 클래스 TeachingMachineApp(1702)은 퀴즈에 기초한 애플리케이션을 위해 런타임 및 사용자 인터페이스를 제공하는 역할을 한다. 바람직하게, 이 애플리케이션은 시험된 영역(examined field)에서 사용자 프로파일(사용자 모델)을 구성하는 역할을 하는 TMEngine(1704)을 내장하고 있다. 예를 들어, 그 일반 영역(general field)이 영어 단어 모음(vocabulary)일 경우, TMEngine(1704)은 단어 관계들, 부정(negation), 기능(function), 토픽 및 그 이상의 것에 관련하여 영어 단어모음의 다양한 서브 영역들에서 사용자의 성공률을 바람직하게 학습한다.

그 애플리케이션에 의해 가르쳐지는 일반 영역의 다양한 서브 영역들에 있어서의 사용자의 성과 혹은 성적(performance)를 분석한 후에, TMEngine(1704)는 그 성과가 저조했던 토픽들과 서브 영역들에 있어서 사용자의 지식을 테스트하고 향상시키도록 그 애플리케이션에게 바람직하게 지시한다. 바람직하게 TMEngine(1704)은 어느 정도 반복되는 사용자 평가를 바람직하게 행하고, 이어서 교육 및 사용자의 성과에 대한 적응을 행하여, 사용자의 새로운 상태와 관련된 퀴즈 문제들을 생성한다.

TMEngine(1704)는 시간이 경과함에 따라 사용자의 성과를 모으게 되고, 사용자의 성공률과 관련한 통계를 TeachingMachineApp(교육 기계 애플리케이션)(1702)에 임의로 제공할 수 있다.

바람직하게 확장가능한 퀴즈 프레임워크는 추상화 레이어들과 인터페이스들을 이용하는 것에 의해 바람직하게 제공된다. TMEngine(1704)은 바람직하게 퀴즈들의 수용체(container)이고; 그 퀴즈들은 모두 TMQuiz(1706) 표준 인터페이스를 바람직하게 구현하므로 그 퀴즈들은 임의로 심리스(seamless)적일 수 있다. 각 퀴즈는 TMDATAccess(1708) 클래스를 이용하여 문제, 답 및 사용자 성공률에 대한 자신과 관련된 데이터베이스에 대한 엑세스 및 저장을 할 수 있다. 상기 교육 기계의 퀴즈 및 토픽 훈련 요소들(aspects)은 바람직하게 분리되어 있으며, 이에 따라 적응형 교육 애플리케이션이 서로 다른 많은 형태의 토픽들과 함께 동작할 수 있게 되고 확장성이 매우 커지게 된다.

몇 가지 서로 다른 형태의 퀴즈들의 예에는 TMWordNet(TM 워드넷) 퀴즈(1710), TMTriviaQuiz(TM 트리비아 퀴즈)(1712) 및 TMRelationQuiz(TM 관련 퀴즈)(1714) 등이 포함된다.

도 17b는 본 발명에 따른 전형적인 교육 시퀀스 다이어그램을 나타낸 것이다. 애플리케이션 관리자(502)(도 5를 참조하여 더욱 상세히 설명됨)는 스텝을 TeachingMachineApp(1702)에 전송한다(화살표 1). 이어서 TeachingMachineApp(1702)은 그 다음 교육 라운드를 준비하라는 요청을 TMEngine(1702)에게 전송한다(화살표 1.1). 이러한 준비는 그 다음 문제를 TMQuiz(1706)에게 요청함으로써(화살표 1.2.1 및 1.2.1.1) 바람직하게 시작된다. 그에 대한 답변은 사용자로부터 수신되어 TMEngine(1704) 및 TMQuiz(1706)에 의해 평가된다(화살표 1.2.2). 만약 답이 옳다면, TMQuiz(1706)은 정답 카운트를 갱신하고(화살표 1.2.2.1.1.1); 그렇지 않다면 오답 카운트를 갱신하며(화살표 1.2.2.1.2.1); 전체 성공률 또한 갱신한다. 바람직하게 TMEngine(1704)는 그 퀴즈 통계치를 저장한다. 임의로 및 바람직하게, 오답이 선택되었다면 정답을 사용자에게 표시한다.

이전에 적어도 한번 사용자가 테스트되었으면, 그러한 시퀀스의 그 다음 부분이 임의로 실행될 수 있다. 애플리케이션 관리자(502)는 다시 스텝을 전송한다(화살표 2). TeachingMachineApp(1702)은 교육 라운드를 준비하라는 요청을 전송한다(화살표 2.1). 사용자의 가장 약한 토픽이 배치되고(화살표 2.1.1), 사용자가 가장 약한 타입의 퀴즈도 바람직하게 배치된다(화살표 2.1.1.2). 그 라운드의 각 문제에 대해서, TeachingMachineApp(1702)은 위에서처럼 그 다음 문제를 바람직하게 얻고, 상술한 바와 같이 사용자의 답을 평가한다.

이러한 구성은 새로운 토픽들에 대해서 또한 새로운 퀴즈 구조들에 대해서 바람직하게 확장가능하다. 새로운 토픽들에는 (예를 들어 영어와 같은) 일반 토픽과 (예를 들어 미국 속어 혹은 여행 단어) 특정 타입의 콘텐츠 등이 바람직하게 포함된다. 바람직하게 이러한 토픽에는 그 토픽을 위한 데이터와 퀴즈 구조가 포함되어 교육 기계가 그 데이터를 퀴즈 구조와 자동으로 결합시킬 수 있다. 바람직하게 각 퀴즈는 퀴즈 템플릿에 기초하고 있으며, 그 템플릿 내의 특정 위치(들)에 임으로 배치될 수 있는 데이터에 대한 명령(instruction)들을 이용한다.

### 예 3: 지능형 에이전트용 진화 시스템

이 예에서는 본 발명에 따른 진화 시스템의 바람직한 실시예에 대해서 기술하기로 하며, 여기에는 본 발명의 바람직한 실시예에 따른 생물 혹은 아바타를 위한 DNA에 대한 설명과, 본 발명에 따른 임의의(optional) 유전자 스튜디오에 대한 설명이 포함되며, 이것들에 한정되는 것은 아니다. 임의로 및 바람직하게 그러한 진화 시스템은 생물 혹은 아바타가 "진화"할 수 있도록, 즉, 그 생물의 행동 및/또는 모습의 적어도 한 측면을 변경하는 것이 가능하게 한다. 이 예에서는 예 2에서 기술된 지능형 에이전트와 함께 임의로 및 바람직하게 동작하는 것으로서 기술되었으나, 이러한 설명은 단지 예시에 불과하며 어떠한 방식으로도 한정하려고 하는 의도는 없다.

지능형 에이전트의 진화(변화)에 대해서는, 아바타 혹은 생물에 의해 표시되는 그 에이전트의 유형(tangible)의 특징들과 그 아바타 혹은 생물의 행동에 영향을 주는 그 에이전트의 무형의 특징들의 양쪽 모두와 관련하여 설명되어 있다.

도 18a는 전형적인 진화 클래스 다이어그램(1800)을 나타내는 것이다. 이 클래스 다이어그램에서 기술되어 있는 유전 모델은 지능형 에이전트의 기능적 성질뿐만 아니라 시각적 성질을 포함한 지능형 에이전트의 여러 가지 성질들이 변경될 수 있도록 한다. 그러한 모델에는 DNA 구조를 표현하는 CreatureDNA 클래스(1802)가 포함된다. 이 DNA 구조는 이용 가능한 유전자들로 이루어진 벡터이며, 새로운 유전자를 포함시키도록 바람직하게 확장될 수 있다. 유전자(gene)는 소정 범위의 가능한 값들(즉, 유전자형(genotype))을 갖는 파라미터이다. 유전자는 본 발명에 따른 시스템에 의해 해석(interpret)되어, 그 유전자 안의 데이터의 표현(expression)이 그 유전자형이 된다. 예를 들어 머리 유전자는 DNA의 첫 번째 유전자로서 위치하며, 그 값은 그 생물의 머리의 시각적 구조로서 표현되며, 그러나 머리의 색깔은 다른 유전자로 부호화(encoded)되는 것이 바람직하다.

지능형 에이전트를 진화시켜 사용자를 기쁘게 해주는 특정 DNA 인스턴스를 얻어내기 위해서, 본 발명에 따른 유전적 모델은 DNA를 변경하는 혼성 및 돌연변이 유전적 동작(hybrid and mutate genetic operations)을 바람직하게 구현한다. CreatureProxy 클래스(1804)는 상기 시스템을 위한 유전적 동작들에게, 또한 DNA에게, 인터페이스를 제공하는 역할을 한다. 또한 CreatureProxy(1804)는 지능형 에이전트에 대한 다른 비유전적 정보(즉, 이름, 생일 등)를 바람직하게 보유한다.

바람직하게 EvolutionMGR 클래스(1806)는 지능형 에이전트의 진화를 관리하고 지능형 에이전트의 CreatureProxy(1804)에게 인터페이스를 제공하고 그 유전적 동작들을 애플리케이션들에게 제공한다.

EvolutionEngine(진화엔진) 클래스(1808)는 소정의 유전적 동작이 일어나서 지능형 에이전트 DNA에 대해서 행해져야 한다는 것을 지시내기 위해, 때때로 생성될 수 있는 진화 이벤트에 주위를 기울인다. 그 DNA 구조는 아래와 같이 주어진다.

바람직하게 CreatureDNA(1802)는 EvolutionEvent(1810)로부터 그러한 진화 이벤트에 주위를 기울인다.

### DNA 구조

```
#ifndef __CREATURE_DNA__
#define __CREATURE_DNA__
#include "CreatureDefs.h"
#include "CommSerializable.h"
```

```

#define GENE_COUNT 19

#define BASE_COLOR_GENE 8

typedef struct internal_dna
{
    unsigned char head;
    unsigned char head_color;
    unsigned char head_scale;
    unsigned char body;
    unsigned char body_color;
    unsigned char body_scale;
    unsigned char hand;
    unsigned char hand_color;
    unsigned char hand_scale;
    unsigned char tail;
    unsigned char tail_color;
    unsigned char tail_scale;
    unsigned char leg;
    unsigned char leg_color;
    unsigned char leg_scale;
    unsigned char dexterity;
    unsigned char efficiency;
    unsigned char interactive;
    unsigned char base_color;
} internal_dna;

typedef internal_dna p_internalDna;

/***
 * 이 클래스는 생물 DNA 구조를 나타낸다.
 * DNA는 생물 몸체 부분들과 몇몇 성격 및 기능적 특질(personality and functional qualities)에 대한 모든 데이터를 보유한다 */
class CreatureDNA /*: public CommSerializable*/
{
public:

```

```

static const int gene_count;

/***
 * 디폴트 구성자(constructor), DNA는 0으로 초기화됨
 */
CreatureDNA();

/***
 * 카피 구성자(Copy constructor)
 * @param other - the DNA to copy
 */
CreatureDNA(const CreatureDNA &other);

/***
 * 초기화 기능, 구성자가 호출되지 않았다면 호출되어야함.
 */
void init();

/***
 * DNA 데이터를 랜덤화함
 */
void randomizeDna();

/***
 * DNA 실제 데이터
 */
union {
    internal_dna genes;
    unsigned char data[GENE_COUNT];
};

/***
 * 형태 유전자의 범위(Range of type gene)
 */
static const int TYPE_RANGE;

```

```

* 색깔 유전자의 범위(Range of color gene)
*/
static const int COLOR_RANGE;
/***
* 스케일 유전자의 범위(Range of scale gene)
*/
static const int SCALE_RANGE;
/***
* 캐릭터 유전자의 범위(Range of character genes)
*/
static const int CHARACTER_RANGE;
static const int BASE_COLOR_RANGE;
private:
/***
* 형태, 색깔, 스케일 트리플렛의 위치(Location of scale gene in the type,color,scale triplet)
*/
static const int SCALE_LOCATION;
}

#endif /*_CREATURE_DNA_*/

```

지능형 에이전트 DNA 구성은 바람직하게 다음과 같이 행해진다. 생존형 이동 전화(living mobile phone)의 버전을 제공하면, DNA는 지능형 에이전트의 각 구성 요소(Building Block)를 위한 Gene으로부터 바람직하게 구성되어진다. 임의로, 그 구성 요소는 그 지능형 에이전트의 시각적 부분일 수 있으며, 여기에는 색깔 혹은 스케일(그 구성요소의 크기)이 바람직하게 포함되며, 또한 지능형 에이전트의 기능 및 행동과 관련된 비시각적 성질도 바람직하게 포함된다. DNA 구성의 그러한 모델은 더 많은 구성요소들이 추가될 수 있고 각 구성 요소의 발현 수준이 증가할 수 있기 때문에, 확장될 수 있다.

DNA 구조로부터 지능형 에이전트를 구성하는 것은 각 유전자와 그 값과 관련하여 행해지는 것이 바람직하다. 각 유전자(building block) 값(발현 수준)은 구성된 에이전트에서 표현된 다른 유전자형을 기술한다. 시각적 에이전트의 기초 구성 요소들(basic building blocks)은 원형들(prototypes)로서 모델화되며, 따라서 원형들의 양은 각 시각적 유전자의 범위를 결정(dictate)하게 된다. 또한, 실행 중에 원형들에 의존하지 않는 발현 유전자들의 값들을 생성할 수 있고, 예를 들어 색깔 유전자 발현 수준은 호스트 플랫폼 색깔 테이블에서의 인덱스들로서 연산될 수 있거나, 혹은 스케일도 호스트 스크린 크기에 대해서 연산될 수 있어서, 미리 결정되어 있는 원형으로부터 독립적인 유전자형을 얻을 수 있게 된다. 그 원형 모델들은 바람직하게 분해(decomposed)되고, 이어서 비원형 에이전트(non-prototype agent)는 각 구성요소의 유전자 값들에 따라 바람직하게 재구성(recomposed)된다.

다음과 기술하는 예는 이러한 과정에 대한 예시적인 비 한정적 설명을 하기 위한 것이다. 단순화 및 명확화를 위해, 색깔 및 스케일 및 다른 비 시각적 유전자들이 포함되지 않았으며, 이러한 유전자들에게 동일한 과정이 적용되게 된다.

DNA의 16-원형 및 5-구성요소 버전이 다음과 같이 임의로 주어질 수 있다:

$$\text{DNA}_0 = \{[\text{head}, 0:15], [\text{body}, 0:15], [\text{legs}, 0:15], [\text{hands}, 0:15], [\text{tail}, 0:15]\}$$

5개의 구성요소들은 각각 원형 모델들의 개수로부터 도출된 구성 요소 유전자 값들에 따라 16개의 서로 다른 가능한 유전자형들을 가지고 있다. 지능형 에이전트를 구성할 때, 그 각 유전자의 값인 DNA내의 구성요소의 값에 따라 오른쪽 구성요소가 취해지게 된다.

예를 들어 상술한 DNA 방식의 특정한 인스턴스는 다음과 같을 수 있다:

DNA = { [3],[5],[10],[13],[0] }

이러한 간단한 DNA 버전에 있어서의 가능한 지능형 에이전트 구성들의 다양도(variety)는 다음과 같다:

$$V_0 = (16) * (16) * (16) * (16) * (16) = (16)^5 = 1048576$$

지능형 에이전트의 일반적 색깔을 기술하기 위한 기본 색깔 유전자(즉, 녹색, 청색 등)가 추가되면, 가능한 16개의 기본 색깔의 발현수준을 가지는 경우, 다음과 같은 다양도가 얻어지게 된다:

DNA<sub>1</sub> =

{[head,0:15], [body,0:15], [legs, 0:15], [hands,0:15], [tail, 0:15], [bs\_color,0:15]}

그러면 다양도는 다음과 같이 된다:

$$V_1 = V_0 * 16 = (16)^6 = 16777216$$

(밝은 색부터 어두운 색깔까지의) 기본 색깔 유전자를 위한 강도 유전자(intensity gene)가 이 DNA 버전에 추가되면, 기본 색깔의 가능한 16개의 강도들의 발현수준을 가지는 경우, 다음과 같은 다양도가 바람직하게 얻어진다:

DNA<sub>2</sub> =

{[head,0:15], [body,0:15], [legs, 0:15], [hands,0:15], [tail, 0:15], [bs\_color,0:15], [intensity,0:15]}

이 경우 다양도의 계산은 다음과 같다:

$$V_2 = V_1 * 16 = (16)^7 = 268435456$$

돌연변이 시퀀스 다이어그램과 혼성화(hybridization) 시퀀스 다이어그램을 각각 나타내고 있는 도 18b 및 도 18c를 참조하여 기술되는 바와 같이, 다양한 유전적 동작들(genetic operations)이 DNA 상에서 임의로 수행될 수 있다.

도 18b에 나타낸 바와 같이, 돌연변이 될 수 있는 유전자 세트(임의로 전체 DNA일 수 있음)로부터 소정의 유전자가 기본 돌연변이 동작에 의해 랜덤하게 선택되어지며, 그 유전자의 가능한 범위(발현수준들) 내에서 선택된 유전자의 값이 변경되게 된다. 그 기본 동작은 여러 번 임의로 수행될 수 있다.

돌연변이 애플리케이션(1812)은 돌연변이체(mutant)를 생성하라는 요청을 EvolutionMGR(진화 MGR)(1806)(화살표 1.1)에게 전송한다(화살표 1.1). EvolutionMGR 클래스(1806)는, 임의로 다수의 돌연변이체를 위해, 이 요청을 CreatureProxy(생물 프락시)(1804)에게 전달한다(그 값은 함수 호출(function call)에 있어서 주어질 수 있다(화살표 1.1.1)). 그러한 돌연변이체 각각에 대해서, CreatureProxy(1804)는 랜덤 유전자를 바람직하게 선택하고(화살표 1.1.1.1.1), 그것을 상기 유전자의 범위 내에 여전히 있는 값으로 변경하게 된다(화살표 1.1.1.1.2). 그 돌연변이체(들)는 돌연변이 애플리케이션(1812)으로 다시 전달되고, 바람직하게 사용자에게 표시되며, 이것에 대해서는 아래의 예 4에서 더욱 상세히 설명하기로 한다.

사용자가 돌연변이체를 승인하면, 돌연변이 애플리케이션(1812)은 그 에이전트의 기존의 구현체(existing implementation)를 그 새로운 돌연변이체(화살표 2.1)로 대체하라는 명령을 EvolutionMGR(1806)에게 전송한다. 그러면 EvolutionMGR(1806)은 CreatureProxy(1804)에서의 그 생물을 위한 DNA를 설정하고(화살표 2.1.1), 이어서 CreatureProxy(1804)는 Agent\_history(에이전트\_히스토리)(1814)에서의 그 에이전트의 히스토리를 바람직하게 갱신한다(2.1.1.1).

도 18c는 2개의 후보 DNA가 서로 배열될 때 발생하는 기본 혼성 동작(basis hybrid operation)(혹은 교차 동작(cross-over operation))을 위한 전형적인 시퀀스 다이어그램을 나타낸 것이다. DNA 상에 위치하는 하나 이상의 교차 포인트들이 바람직하게 선택된다(교차 포인트 개수는 1부터 DNA 내의 유전자 개수까지 변경될 수 있고; 임의로 이 개수는 랜덤하게 선택될 수 있음). 교차 포인트들을 선택하는 동작은 get\_cut\_index라고 불린다. 각 교차 포인트에서, DNA를 위한 값이 기존의 DNA 값들 중 하나로부터 선택된다. 임의로, 이것은 cutting\_index라고 불리는 카운트에 따라 랜덤하게 이루어질 수 있다. 그 결과 상기 2개의 후보 DNA가 서로 혼합된다. 임의로 그러한 기본 혼성 동작은 여러 후보들을 이용하여 여러 번에 걸쳐 수행될 수 있다.

도시된 바와 같이, HybridApp(1816)는 혼성화 과정을 시작하라는 커맨드를 EvolutionMGR(1806)에 전송한다. 임의로, 이 혼성화 과정은 사용자가 그 혼성 에이전트를 승인하거나 그 과정을 중단할 때까지 수행된다. EvolutionMGR(1806)은 수행될 다수의 교차(혼성화)들을 이용하여, CreatureMGR(1806)로부터 타겟 DNA(화살표 2.1.1)를 획득하라는 명령을 전송함으로써 혼성화를 시작시키게 된다. 도시된 바와 같이, cutting\_index는 상기 2개의 DNA의 값들 사이의 교차를 언제 행할지를 지시하도록 유지된다.

상기 혼성 에이전트는 복귀(return)되어지며, 사용자가 동의하면, 현재 에이전트는 돌연변이 과정과 관련하여 상술한 바와 같이 그 혼성 에이전트로 대체된다. 결과적으로 agent\_history(1814)에서의 그 에이전트의 히스토리가 바람직하게 갱신되게 된다.

임의로 및 바람직하게 혼성화는 적외선, 블루투스 혹은 인터넷을 통해, 예를 들어 SMS 메시지에 있는, 이동 정보 장치의 외부에 있는 소스 혹은 그 이외의 임의의 소스로부터 전송된 에이전트 DNA를 이용하여 수행될 수 있다. 단지 설명하기 위한 목적으로 한정하고자 하는 의도 없이, SMS 메시지를 통해 그러한 혼성 DNA를 수신하는 것과 관련하여 그러한 과정을 예시하기로 한다. SMS 메시지는 MIME 타입의 DNA를 위한 데이터를 바람직하게 포함하고 있다. 더욱 바람직하게는, 본 발명의 시스템은 MIME 타입을 위한 후크를 가지고 있어서, 그러한 타입의 SMS 메시지가 사용자에 의한 수동 조정(manual intervention)을 필요로 하지 않으면서 혼성화를 위해 자동으로 바람직하게 파싱된다.

도 19는 그러한 과정의 전형적인 시퀀스 다이어그램을 나타내고 있다. 도시된 바와 같이, 사용자 1은 핸드셋 1을 통해 사용자 1의 지능형 에이전트를 사용자 2의 지능형 에이전트와 혼성화하라는 요청을 전송한다. 사용자 2는 핸드셋 2를 통해 그 요청을 임의로 승인 혹은 거절할 수 있다. 사용자 2가 승인하면 핸드셋 1 상에서 양쪽의 에이전트로부터의 DNA 사이에서 혼성 처리가 행해진다. 그 결과가 요청한 쪽(사용자 1)에게 임의로 표시되며, 이 요청측은 그 혼성을 현재 에이전트를 대체하는 것으로서 저장할 수 있다. 그 혼성이 대체하는 것으로서 사용되면, 사용자 2는 핸드셋 2 상에서 혼성 결과 정보 수집부(hybrid results collection)에 그 혼성에 대한 것을 저장한다.

#### 예 4: 본 발명과 사용자의 상호작용(USER INTERACTIONS WITH THE PRESENT INVENTION)

이 예는 사용자와 상호작용을 하는 본 발명에 따른 시스템의 임의의 그러나 바람직한 실시 예를 설명하기 위한 것으로서, 대표적이고 비한정적이며 예시적인 복수의 화면출력 화상들(screenshots)을 참조하여 기술될 것이다.

도 20은 생물 혹은 아바타(지능형 에이전트의 시작적 표현)인 "부유 에이전트"의 전형적인 화면출력 화상을 나타낸 것이다. 도 21은 지능형 에이전트의 가상 세계를 위한 객체들을 선택하기 위한 메뉴의 전형적인 화면출력 화상을 나타낸 것이다.

도 22는 사용자 옵션(preferences)뿐만 아니라 에이전트 설정을 사용자가 구성 및 변경할 수 있도록 하는 시작 마법사 애플리케이션(Start Wizard application)을 나타낸 것이다.

상기 마법사로 수행되는 행위의 예로서, 지능형 에이전트의 감정형 시스템을 위한 설정을 결정, 즉 "개성 결정(Set Personality)"하는 것을 들 수 있다. 이때, 사용자는 생물의 개성 및 경향을 구성할 수 있다.

사용자는 열의(Enthusiasm), 사교성(Sociability), 비사교적 행동(Anti-social behavior), 성질(Temper)(인내 수준), 우울(Melancholy), 이기적 행동 등의 여러 가지 성격들(characteristics)의 수준을 증가시키고, 그 반대를 행하여 그 수준을 감소시키기 위해 오른쪽 화살표 키를 눌러서 그 생물의 설정을 임의로 및 바람직하게 결정할 수 있다.

사용자는, 예를 들어 도움을 얼마나 빨리 받을지를 결정하기 위해, 사용자 옵션을 바람직하게 설정할 수 있다. 이들 옵션들 중 몇 가지의 비한정적 예에는, 통신(그 에이전트가 통신하는 정도); entertain\_user(사용자를\_즐겁게\_하기)(사용자와 노는 에이전트를 제어); entertain\_self(자신을\_즐겁게\_하기)(혼자 노는 에이전트를 제어); preserve\_battery(배터리\_보존)(배터리 수명 늘리기); transparency\_level(투명도\_수준)(생물의 투명도 수준) 등이 포함된다.

또한, 사용자는 그 시작 마법사로 사용자 신상정보(User Details)를 바람직하게 설정하며, 그 사용자 신상정보에는 사용자 이름, 생일 및 성별 등이 포함되며, 이것들에 한정되는 것은 아니다. 본 발명의 임의의 실시예에 따르면, 상기 생일 값은 사용자들 사이의 궁합 가능성(konghup possibility)(좋은 점술 패턴(favorable astrology pattern)을 갖는 혼성(hybrid)을 생성할 수 있는 능력을 말하는 것으로, 궁합 옵션(konghup option)은 날짜들 및 별점들(horoscopes)에 대한 적절한 테이블에 따라 구성되어짐)을 결정할 것이므로 혼성 SMS에 있어서 중요하다.

또한, 사용자는 생물의 신상정보(Creature Details)도 바람직하게 설정할 수 있다.

도 23은 상술한 바와 같은 혼성 애플리케이션을 통하여 혼성화를 수행하기 위한 전형적인 메뉴를 나타낸 것이다.

도 24a는 사용자가 생물을 랜덤하게 생성할 수 있도록 하는 "생성(Generate)" 버튼을 누름으로써, 새로운 생물을 보기 위한 (또한 임의로 다시 생성하기 위한) 전형적인 화면출력 화상을 나타낸 것이다. 도 24b는 "혼성(Hybrid)" 버튼을 눌러서 얻어진 화면출력상의 생물을 나타낸 것으로, 그 버튼을 누르면 사용자의 생물 선택을 확정하게 되고 생물 미리보기 창으로 이동하게 된다.

이 미리보기 창에 의해 사용자는 새로 생성된 생물을 3차원으로 볼 수 있게 되고, 임의로 다음과 같은 옵션들을 사용하여 그 생물을 동화/animate)시킨다.

1. 위쪽 방향키(Navigation UP key): 줌인(Zoom In)하고, 생물의 크기를 최소화시킴.
2. 아래쪽 방향키(Navigation DOWN key): 줌아웃하고, 생물의 크기를 최대화시킴.
3. 왼쪽 방향키(Navigation LEFT key): "확인(OK)" 버튼과 "뒤로 가기(Back)" 버튼 사이의 전환.

4. 오른쪽 방향키(Navigation RIGHT key): "확인" 버튼과 "뒤로 가기" 버튼 사이의 전환.
5. 확인키(OK): 선택을 확인.
6. 삭제키(CLR): 생물 미리 보기 창에서 생존형 이동전화 메뉴(Living Mobile Menu)로 나가기.
7. 종료키(End): 생물 미리 보기 창에서 메인 메뉴로 나가기.
8. '0' 키: 생물에 대해서 밝게 및 어둡게 하는 동작을 행하기.
9. '1' 키: 시계방향으로 생물을 왼쪽으로 회전시킴.
10. '2' 키: 생물을 3차원으로 회전시키기.
11. '3' 키: 시계 반대 방향으로 생물을 오른쪽으로 회전시킴
12. '5' 키: 생물을 3차원으로 회전시키기
13. '6' 키: 생물에게 여러 가지 방식으로 동작을 부여한다. 이 키를 누를 때마다 동화 형태(animation type)가 바뀐다.

상기 생물이 행할 수 있는 동화 형태에는 걷기, 앓기, 냄새 맡기, 날기, 점프하기 등이 임의로 포함되며, 이것들에 한정되는 것은 아니다.

도 25는 사용자로 하여금 그 세대(generations)에서의 동물의 변화에 대한 히스토리를 살펴보고 검사할 수 있도록 하는 혼성 히스토리의 전형적인 출력화면 화상을 나타낸 것이다. 사용자는 현재의 생물과 모체(parents)를 바람직하게 볼 수 있고, 또한 임의로 그 모체의 모체도 볼 수 있다. 바람직하게, 모든 생물에 대해서, 많아야 2개의 모체가 있을 수 있다.

도 26은 현재 생물의 DNA 서열을 가지고 있는 유전자 스튜디오의 전형적인 화면출력을 나타낸 것이다. 또한, 이 유전자 스튜디오는 사용자로 하여금 에이전트의 DNA 서열을 변경 및 수정할 수 있는 기회를 바람직하게 제공한다.

#### 예 5: 네트워크형 이동 정보 장치를 위한 지능형 에이전트

이 예는 네트워크형 이동 정보 장치(networked mobile information device)(바람직하게는 휴대폰)에 대한 지능형 에이전트의 사용에 관한 것이다. 임의로 및 바람직하게, 지능형 에이전트는 사용자와 상호작용하기 위한 아바타와, 다른 이동 정보 장치와 같은 네트워크 상의 다른 구성요소 및/또는 네트워크 자체와 상호작용하는 에이전트를 포함한다. 따라서, 바람직하게 아바타는 사용자 인터페이스(혹은 그 일부)를 형성하고, 모습(더욱 바람직하게 3차원 모양)을 갖게 된다. 그 모습은 임의로 사람의 모습이 될 수 있지만, 그 대신에 다른 타입의 실제 혹은 상상의 캐릭터 또는 생물에 기초한 것일 수 있다. 상기 에이전트는 아바타와 이동 정보 장치와의 사이, 및/또는 네트워크 상의 다른 구성요소들과의 사이, 및/또는 다른 이동 정보 장치들의 다른 아바타들과의 사이 등의 통신을 처리한다. 그 구현에 대해서 휴대폰과 같은 이동 정보 장치를 참조하여 설명되었지만, 그 구현의 아바타 측(또한 에이전트 그 자체도)은 상술한 바와 같은 적응형 시스템(예 2) 및/또는 사전행동적 사용자 인터페이스(예 1)로 임의로 구현될 수도 있다는 것을 유념해야 한다.

본 발명의 지능형 에이전트는 "생존형 장치(Living Device)"의 개념을 적용함으로써 향상된 감정적 경험을 생성하는 것을 목표로 하고 있다. 모든 살아있는 생물은 모습 및 행동에 있어서 유일하고 특별한 것으로서 지능형 에이전트의 유일성(uniqueness)에 중점을 두는 것과, 또한 사용자의 생존형 장치와의 상호작용을 향상시키기 위한 다양한 아바타 모양 등의 다양성을 제공하는 것이, 그러한 개념에 바람직하게 포함된다. 아바타는 주의를 끌만한 (compelling) 화상 특성들을 가지고 있으며, 이것은 적절한 보충적인 객체들과 주변의 환경을 임의로 갖는다.

바람직하게, 지능형 에이전트는 그 자존(self-existence) 및 독립적 학습을 나타내는 예상되지 않은 행동으로, 지능형 의사결정을 표시한다. 그러한 독립적 행동은 본 발명의 중요한 특징 중 하나로서, 어떠한 타입의 사용자 인터페이스 또는 사용자를 위한 상호작용을 위해서도 또한 어떠한 타입을 갖는 연산장치를 위해서도 이전에 보여진 바 없으며, 이동 정보 장치를 위한 지능형 에이전트를 위해서 확실히 이용된 바 없다. 또한, 상술한 지능형 에이전트는 생물처럼 시간에 따라 진화하여 시각적 변화를 표시하게 된다. 이것은 "생존형 장치"의 가장 중요한 특성들 중 하나이다.

진화단계는 놀람 및 다음 진화단계에 대한 기대에 대한 사용자로부터의 감정적 반응을 유발시킨다.

진화는 시간에 따른 생물의 시각적 변화에 해당한다. 시간프레임은, 유통중인 중급 휴대폰의 수명에 해당하는 것으로서 예를 들어 일년으로 임의로 설정될 수 있다. 일년 동안에, 진화를 통해 주기적인 변화가 바람직하게 발생한다. 진화 경로(환경에 대한 적응)는 자연선택의 결과이다. 또 다른 선택옵션으로서 자동 선택을 위해 몇 가지 기능을 개발함으로써 미리 결정되는 자연 선택과정이 있지만, 자연선택은 임의로 사용자 중심(user driven)이 될 수 있다(즉, 사용자는 다음 세대가 더 좋은지 여부를 결정함).

지능형 에이전트는 서로 다른 실시예들의 2가지 "세계"(즉, 전화기 세계 및 가상 생물 세계)에서 기능하기 위해 임의로 구현될 수 있다. 전화기(이동 정보 장치) 세계에서는, 지능형 에이전트가 그 전화기의 서로 다른 기능들을 제어하여, 상술한 바와 같이, 사용자에게 여러 가지 기능의 선택들을 제안할 수 있게 된다. 바람직하게, 지능형 에이전트는 후속하는 그 에이전트를 위해 모델이 되는 하나 이상의 전화기 사용 처리과정들에 기초하여 동작할 수 있다. 전화기 세계의 또 다른 중요한 특징은, 화면 브레이킹(screen breaking)과 같은 그래픽 표현, 프리 드로잉(free drawing), 혹은 얼굴(표면) 또는 문자 표현(특정한 경우에 하나 또는 두 개의 관련 단어)일 수 있는 감정표현들에 있다.

가상 세계는 바람직하게 화상 표시 및 활동 영역(playground area)이며, 아바타 이외의 객체들이 삽입될 수 있고 아바타 학습 및 그들과의 상호작용을 사용자가 관찰할 수 있는 영역이 임의로 될 수 있다. 그 세계에 진입할 수 있는 그러한 객체들은 임의로 미리 결정될 수 있고, 그 학습과정으로부터 생긴 가능한 서로 다른 행동들을 갖게 된다. 임의로 및 바람직하게, 사용자는 보상들 또는 행동억제요인(disincentive)들을 제공할 수 있고, 학습과정의 일부가 될 수 있다. 이러한 관점에서, (아바타의 모습을 통해) 지능형 에이전트는 가상 애완동물(pet) 또는 동료(친구)의 한 형태로서 임의로 행동할 수 있다.

지능형 에이전트의 몇 가지 바람직한 형태들에는, (아바타의 모습과 관련한) 3D 그래픽 기반구조와; 적응형 행동 및 사전행동적 행동 모두를 지원하는 인공지능 및 기계 학습 메커니즘의 사용과; 게임 능력의 제공과; 이동 정보 장치의 유용성을 향상시키고 또한 특정한 사용자의 보조를 제공하는 (그 사용자를 도와주는) 능력과; 호스트 플랫폼의 추상화 레이어(abstraction layer)의 제공 등이 포함되며, 그러나 이것들에 한정되는 것은 아니다. 그와 함께, 이들 기능들은 이동 정보 장치 상에서 동작하도록 모두가 일반적으로 정의된 복수의 인공지능 애플리케이션들을 지원하기 위한 안정적이고, 주목할 만하며, 혁신적인 콘텐츠 플랫폼을 제공한다.

또한, 바람직하게, 아바타는 많은 시각적 요소들을 가지고 있다. 예를 들어 외곽 클립 크기는 60 x 70 픽셀까지 임의로 될 수 있으며, 물론 이동 정보 장치의 화면 표시의 특성들에 따라 다른 해상도가 선택될 수도 있다. 바람직하게, 아바타는 몇 가지 색깔을 갖는 3D 다각형 객체로서 표현되며, 어떠한 경우에도 바람직하게 그림자, 텍스처, 및 애니메이션 지원 등의 몇 가지 색깔을 갖는 복수의 서로 다른 3D 화상 특성들을 지닌다. 이러한 능력들은 이동 정보 장치에 저장되어 있는 이전에 생성된 시각적 구성 요소들(visual building blocks)을 통해 임의로 제공될 수 있다. 바람직하게, 아바타의 시각적 모습은 실행시 이루어진다.

아바타는 사용자의 성향을 고려하여, 시작 마법사(launch wizard) 이후에 "생존하기(living)"을 임의로 시작시킬 수 있다(생존형 장치에 사용자 소개하기). 진화와 더불어, 아바타는 돌연변이를 나타내는 작은 시각적 변화들을 임의로 표시할 수 있다(임의의 단계(random step)에서 몇 개의 키 버텍스(key vertices)의 이동/색깔 변경). 시각적 진화단계는 하나의 구성 요소를 추가/대체함으로써 수행되는 것이 바람직하다. 바람직하게 아바타는 모든 방향으로 이동할 수 있고, 완전히 살아 움직이는 3D 캐릭터가 되는 것이 더욱 바람직하다.

아바타는 배경에 있어서 이동 정보 장치 사용자 인터페이스를 갖는 이동 정보 장치 위에 떠 있는 것처럼 보이는 것이 바람직하며, 사용자의 요청에 따라 퇴장될 수도 있다. 아바타는 현재 사용자의 이동 정보 장치와의 정상적(normal)인 상호작용을 이용할 수 있어서, 사용자에 의한 강제적 숨김/퇴장을 최소화하려고 하는 것이 바람직하다.

본 발명의 임의의 그러나 바람직한 실시예들에 따르면, 아바타는 더욱 자연적이고 물리적으로 현실적인(realistic) 방식으로 화면상에서 "이동"하도록 프로그램될 수 있다. 예를 들어, 로봇의 이동을 제어하기 위한 물리적으로 현실적인 행동 및 이동을 기술하려고 시도하는 여러 가지 알고리즘 및 변수들이 이용 가능하다. 그러한 알고리즘들 및 변수들의 예는 "Automatic Generation of Kinematic Models for the Conversion of Human Motion Capture Data into Humanoid Robot Motion"(Ude et al., Proc. First IEEE-RAS Int. Conf. Humanoid Robots (Humanoids 2000), Cambridge, MA, USA, 2000년 9월)에 기술되어 있다(이 참고문헌은 그 내용이 본 명세서에 완전히 기재되어 있는 것과 같은 효과를 갖도록 본 명세서와 함께 제출함). 이 참고문헌은 여러 가지 인간 동작 촬영 기술들과, 촬영된 데이터를 인간형 로봇 동적 변수(kinetic parameters)로 자동 변역하는 방법에 대해서 설명하고 있다. 그 내용을 요약하면, 인간 및 로봇 동작이 모델이 되며, 그 모델들은 실제 인간 동작 데이터를 인간형 로봇의 동작을 제어하는데 이용될 수 있는 데이터로 변역하기 위해 이용된다.

이러한 종류의 참고자료는 인간형 로봇의 이동의 모델을 얻는 방법에 대한 정보를 제공하기 때문에 유용하다. 본 발명은 아바타(3차원적으로 묘사된 가상 캐릭터)의 실제적인 이동과 관련되지만, 그와 유사한 모델들을 인간형 로봇을 위한 것으로서 아바타를 위해 임의로 이용될 수 있을 것이다. 또한, 임의로, 하나의 모델을 동물의 이동들을 모델화하기 위해 구성하여 동물 또는 동물 형태의 아바타의 더욱 실제적(현실적) 이동을 가능하게 할 수도 있다. 더욱 일반적으로, 상기 시스템은 주어진 어떠한 3D 캐릭터 데이터도 일반적(generalized)으로 취급할 수 있는 것이 바람직하다.

임의로 및 바람직하게, 이들 모델의 변수들이 진화 과정동안에 임의로 변화되어 아바타가 움직이는 방식을 변경할 수 있으므로, 아바타의 움직임이 진화될 수 있도록 하기 위해 그 모델들을 임의로 및 바람직하게 이용할 수도 있다. 또한, 바람직하게, 그러한 모델들은 아바타의 비결정적 움직임을 기술하기에 유용하며, 또한 임의로 비결정적 움직임들이 진화될 수 있도록 하기 위해 유용하다. 그러한 비결정적 행동은 또한 사용자의 흥미를 유지하는데 도움을 준다.

또한, 본 발명의 다른 바람직한 실시예들에 따르면, 아바타의 행동은 비결정적 모델에 따라 임의로 및 바람직하게 생성되고 관리되어진다. 그러한 모델들은 "A Behavior Language for Story-Based Believable Agents"(M. Mateas and A. Stern, Working Notes of Artificial Intelligence and Interactive Entertainment, AAAI Spring Symposium Series, AAAI Press, USA, 2002)에 기재되어 있는 것처럼, ABL(A Behavioral Language)와 같은 공지된 행동 언어로 임의로 쓰여질 수 있다(이 참고문헌은 그 내용이 본 명세서에 완전히 기재되어 있는 것과 같은 효과를 갖도록 본 명세서와 함께 제출함). 이 참고문헌은 실제와 같은 방식으로 행동하는 가상 캐릭터들을 생성하는데 이용될 수 있는 ABL에 대해서 논하고 있다. 그러한 현실적 행동들에는, 예를 들어 음성, 또한 움직임 및/또는 제스처를 통해, 입력에 대해서 반응하는 것이 포함되며, 이들 모두는 소프트웨어의 인간 사용자와 현실적인 통신을 제공

하는 것이다. "움직임"이란, 모습에 있어서 현실적인 움직임의 모델화(modeling)를 반드시 의미하는 것은 아니며, 그 정황에 있어서 현실적인 움직임을 그 발생시에 지원하는 것을 의미한다는 점을 유념해야 한다.

여러 가지 입력 및 출력을 포함하며, 인간 사용자와 현실적이고 비결정적인 상호작용적 행동을 모델화 및 지원하기 위해 이용될 수 있는 그러한 언어는 본 발명에 따른 아바타의 행동을 위해 임의로 이용될 수 있다. 예를 들어, 그 언어는 표현적인 행동의 예로서의 "비트 이디엄(beat idioms)"를 기술하게 된다. 이들 비트 이디엄은 비트 목적들, 핸들러들 및 교차-비트 행동들의 세 가지 카테고리로 분류된다. 비트 목적(Beat goals)은 인간 사용자에게 인사하는 것과 같이, 특정한 상황과의 관계에 있어서 행해져야 하는 행동들을 말한다. 핸들러들(Handlers)은 인간 사용자와 가상 생물(예를 들어 본 발명의 아바타) 사이의 상호작용, 또는 가상 생물들 사이의 상호작용을 담당한다. 교차-비트 행동들(Cross-beat behaviors)은 가상 생물체가 일련의 행동들 또는 비트 이디엄들 사이에서 이동할 수 있도록 한다. 명확히, 그 언어 내의 그러한 구성들은 본 발명에 따라 아바타를 위해 임의로 이용될 수 있다.

물론, ABL은 믿을 만한 에이전트 언어들 중 하나의 비한정적 예에 불과하며, 다른 타입의 언어 및/또는 모델들을 ABL 대신에 및/또는 ABL과 결합하여 임의로 이용될 수 있다는 점을 유념해야 한다.

또한, 아바타는 예를 들어 행복함, 슬픔, 놀람, 유감, 아픔, 또는 지루함과 같은 여러 가지 감정적 표현들을 갖는 것이 바람직하며, 이것은 얼굴 표현일 필요는 없으며 대신에 애니메이션 또는 텍스트일 수 있다. 감정적 표현들은 서로 결합될 수도 있다.

또한, 아바타는 화면의 모양을 변경하고, 사용자에게 글을 쓰고 및/또는 전화기를 통해 소리를 울리는 것 같이 보일 수도 있을 것이며, 이것들은 지능형 에이전트의 동작을 통해 바람직하게 성취되게 된다. 그러한 에이전트는, 예를 들어 아바타가 가상 세계상의 딱딱한 물체(객체)에 부딪쳤을 때, 또는 사용자의 주위를 끌고자 할 때, 진동모드를 임으로 동작시킬 수도 있다. 또한, 임의로 아바타는 전화기의 사용자 인터페이스 화면을 능동적으로 조작하는 듯 보일 수도 있을 듯 보일 수도 있다.

아바타 및/또는 지능형 에이전트의 이러한 서로 다른 기능들을 구현하기 위해서, 도 27 내지 도 30을 참조하여 이하 설명하는 바와 같이 지능형 에이전트를 임의로 및 바람직하게 구성할 수 있으며, 이때 이들 도면은 하나의 구현 예를 나타내는 것에 불과하며 다른 많은 구현 예들이 가능하다는 점을 유념해야 한다. 또한, 지능형 에이전트의 구현은 다시 상술한 예 1 및 예 2에서 언급된 구현 예들을 임의로 결합하거나 그것에 의존할 수 있다.

도 27은 본 발명에 따른 지능형 에이전트 시스템(2700)을 나타낸 개략적인 블록도이다. 도시된 바와 같이, 제1 사용자(2702)는 제2 이동 정보 장치(2704)를 제어하며, 이 장치(2704)는, 단지 설명하기 위한 목적으로 또한 어떠한 한정할 의도도 없이, 이 예의 목적을 위해, 휴대폰으로서 임의로 구현될 수 있다. 제2 사용자(2706)는 제2 이동 정보 장치(2708)를 제어한다. 제1 이동 정보 장치(2704) 및 제2 이동 정보 장치(2708)는 예를 들어 메시지 전송에 의해 네트워크(2710)를 통해 바람직하게 통신한다.

바람직하게, 제1 이동 정보 장치(2704)와 제2 이동 정보 장치(2708)는 각각 그 사용자(2720, 2706)와의 상호작용을 위해서 또한 다른 지능형 에이전트와의 상호작용을 위해서 지능형 에이전트를 특징적 요소로서 가지고 있다. 따라서, 도시된 바와 같이 시스템(2700)에 의해, 하나의 커뮤니티를 이루는 그러한 지능형 에이전트들이 서로 상호작용할 수 있게 되며, 및/또는 예를 들어, 네트워크(2710)를 통해 그들 각각의 사용자들을 위한 정보를 획득할 수 있게 된다.

그들 각각의 이동 정보 장치(2704, 2708)와 사용자들(2702, 2706) 간의 상호작용에는, 바람직하게, 이동 정보 장치의 일반 동작이 포함되며, 또한 "생존형 이동 장치"의 흥미로운 새로운 기능들이 추가된다. 이들 기능들에는, 바람직하게, 지능형 에이전트가 포함되며, 또한 사용자 인터페이스를 제공하고 또한 바람직하게 향상된 사용자의 감정적 경험을 제공하기 위한 아바타의 이용도 포함된다.

이러한 지능형 에이전트는 바람직하게 "지각하고 있는(aware)" 지능형 소프트웨어 프레임워크를 특징적 요소로서 갖는다. 그러한 시스템의 내부 동작은 여러 가지 알고리즘 툴과 관련되어 있는데, 이것에는 AI(인공지능) 및 ML(기계 학습) 알고리즘이 포함되지만 이것들에 한정되는 것은 아니다.

시스템(2700)은 도시된 여러 사용자들 사이의 상호작용에 임의로 관여할 수 있다. 그러한 상호작용에 의해 그 앤드 유저를 위한 이동 정보 장치를 이용하는 유용성 및 즐거움이 증가된다.

도 28은 도 27의 지능형 에이전트 시스템을 좀더 구체적으로 나타낸 것이다. 도시된 바와 같이, 제1 지능형 에이전트(2800)는 행동을 취하고, 학습하고, 이동 정보 장치의 동작에 대해 의사결정을 할 수 있기 위해, 임의로 및 바람직하게, 상술한 지식 베이스와 같은 시나리오 데이터(2802)에 따라 동작할 수 있다. 제2 지능형 에이전트(2800)의 학습 및 발전 과정은 상술한 바와 같이 전화를 위한 전화모듈(2804)에 의해 바람직하게 지원되게 된다. 본 발의 바람직한 실시예에 따라, 제1 지능형 에이전트(2800)가 사용자와 아바타를 통해 통신을 하면, 애니메이션 모듈(2806)은 아바타의 모습을 바람직하게 지원하게 된다.

제1 지능형 에이전트(2800)는, 예를 들어 사용자를 위한 정보를 얻기 위해, 후단 서버(backend server)(2808) 및/또는 컴퓨터(2810)와 같은 다른 네트워크 자원과 네트워크(도시하지 않음)를 통해 임의로 통신할 수도 있다.

제1 지능형 에이전트(2800)는 도시된 바와 같이 제2 지능형 에이전트(2812)와 임의로 통신을 할 수도 있다.

도 29는 본 발명에 따른 행위 선택 시스템(2900)의 구현 예를 나타낸 개략 블록도로서, 이 시스템(2900)은 지능형 에이전트가 행위를 선택할 수 있도록 하는 기반구조를 제공한다.

행위 선택 시스템(2900)은 ActionManager(2902)(설명을 위해 도 10을 참조)를 특징적 요소로서 갖는 것이 바람직 하며, 이것이 행위를 실제로 행하게 된다. 베이스액션 인터페이스(BaseAction interface)(2904)는 ActionManager(2902)에 의해 실행되는 모든 행위를 위한 인터페이스를 제공하는 것이 바람직하다.

행위들은 특정 행위를 수행하기 위해 필요한 AnimationManager(애니메이션 관리자)(2906)와 SoundManager(사운드 관리자)(2908)로서 표시되는 장치 및 애플리케이션 기능들을 임의로 이용할 수 있다. 임의로 및 바람직하게, 각 행위는 정확한 올바른 수행을 위해 적절한 관리자들을 모으게 된다.

임의로 및 바람직하게, AnimationManager(2906)은 ChangeUIAction(UI 행위 변경부)(2910)를 또한 제어할 수 있고, 이것은 사용자 인터페이스의 시각적 표시의 모습(형태)을 변경하게 된다. 추가적으로 또는 그 대신에, 아바타가 지능형 에이전트를 사용자에게 표현하기 위해 이용되면, AnimationManager(2906)는 GoAwayFromObjectAction(객체 행위로부터 벗어나기)(2912) 및 GoTowardObjectAction(객체 행위로 진행)(2914)을 임의로 및 바람직하게 제어할 수도 있고, 그것에 의해 아바타가 아바타의 가상세계의 가상 객체들과 상호작용할 수 있게 된다.

도 30a 및 30b는 이동 정보 장치의 화면상의 본 발명에 따른 아바타의 화면출력의 전형적인 2개의 비한정적 예를 나타낸 것이다. 도 30a는 아바타와의 상호작용을 통해 벨소리 크기를 조정하기 위한 사용자 인터페이스의 화면출력 예를 나타낸 것이다. 도 30b는 아바타와의 상호작용을 통해 메시지를 수신하기 위한 사용자 인터페이스의 화면출력 예를 나타낸 것이다.

본 발명을 한정된 개수의 실시예들을 이용하여 설명하였지만, 본 발명에 대한 많은 변경, 수정, 및 다른 응용들이 행해질 수 있다.

#### 발명의 효과

본 발명은 현재의 포괄적(generic)이고 비유연적(non-flexible)인 사용자 인터페이스를 유연성이 있고 실제적으로 사용자 친화적인 인터페이스로 변화시키길 뿐만 아니라, 연산장치와의 향상된 사용자 경험 및 상호작용을 제공한다. 또한, 본 발명은 사용자와 상호작용하게 되는 생물 혹은 아바타 형태로 사용자인터페이스를 구성함으로써, 연산장치와의 사용자의 향상된 감정적 경험(emotional experience)을 제공한다. 따라서, 본 발명은 예를 들어 휴대폰과 같은 이동 정보 장치에 "생존형 장치(living device)"의 경험을 부여할 수 있다.

#### (57) 청구항 1.

운영체제를 갖춘 연산장치를 위한 사전행동적 사용자 인터페이스에 있어서,

a) 사용자와 상기 운영체제 사이의 통신을 위한 사용자 인터페이스와;

b) 상기 사용자 인터페이스와 상기 사용자의 상호작용의 적어도 하나의 패턴을 검출하고, 상기 검출된 패턴에 따라 상기 사용자 인터페이스의 적어도 하나의 기능을 사전행동적으로 변경하는 학습모듈을 포함하여 이루어진 사전행동적 사용자 인터페이스.

#### 청구항 2.

제1항에 있어서,

상기 적어도 하나의 패턴은 상기 사용자 인터페이스와 상기 사용자의 적어도 하나의 이전 상호작용에 따라 결정된 패턴과, 미리 결정된 패턴, 또는 그 조합으로 이루어진 군으로부터 선택되는 것을 특징으로 하는 사전행동적 사용자 인터페이스.

#### 청구항 3.

제1항 또는 제2항에 있어서,

상기 사용자 인터페이스는 그래픽 표시부를 특징적 요소로서 가지며, 상기 사용자 인터페이스의 적어도 하나의 기능을 상기 변경하는 것은 상기 그래픽 표시부의 적어도 일부분을 변경하는 것을 포함하는 것을 특징으로 하는 사전행동적 사용자 인터페이스.

#### 청구항 4.

제3항에 있어서,

상기 그래픽 표시부의 적어도 일부분을 상기 변경하는 것은

상기 검출된 패턴에 따라 표시할 메뉴를 선택하는 것과;

상기 메뉴를 표시하는 것을 포함하는 것을 특징으로 하는 사전행동적 사용자 인터페이스.

**청구항 5.**

제4항에 있어서,

상기 메뉴를 선택하는 것은 복수의 메뉴 옵션으로부터 하나의 메뉴를 구성하는 것을 포함하는 것을 특징으로 하는 사전행동적 사용자 인터페이스.

**청구항 6.**

제1항 또는 제2항에 있어서,

상기 사용자 인터페이스는 오디오 표시부를 특징적 요소로서 포함하며, 상기 사용자 인터페이스의 적어도 하나의 기능을 상기 변경하는 것은 상기 연산장치에 의해 생성된 적어도 하나의 가청 사운드를 변경하는 것을 포함하는 것을 특징으로 하는 사전행동적 사용자 인터페이스.

**청구항 7.**

제1항 내지 제6항 중 어느 한항에 있어서,

상기 연산장치는 개인용 컴퓨터와, ATM과, 이동 정보 장치, 혹은 운영체제를 갖춘 소비재로 이루어진 군으로부터 선택되는 것을 특징으로 하는 사전행동적 사용자 인터페이스.

**청구항 8.**

제1항 내지 제7항 중 어느 한항에 있어서,

상기 학습 모듈은 사전행동적 변경으로부터 직접적인 사용자의 선택으로 이어지는 상기 사전행동적 변경의 확률을 최대화하는 것을 특징으로 하는 사전행동적 사용자 인터페이스.

**청구항 9.**

제8항에 있어서,

상기 최대화는 학습 보강을 통해 수행되는 것을 특징으로 하는 사전행동적 사용자 인터페이스.

**청구항 10.**

제9항에 있어서,

상기 학습보강은 반복적(iterative) 학습 과정을 통해 수행되는 것을 특징으로 하는 사전행동적 사용자 인터페이스.

**청구항 11.**

제10항에 있어서,

상기 학습 과정의 반복(iteration)은 각각 상기 변경이 수행된 후에 행해지는 것을 특징으로 하는 사전행동적 사용자 인터페이스.

**청구항 12.**

제1항 내지 제11항 중 어느 한항에 있어서,

상기 사용자 인터페이스의 적어도 하나의 기능을 상기 사전행동적으로 변경하는 것은 상기 운영체제를 통해 부가 소프트웨어 애플리케이션을 작동시키는 것을 포함하는 것을 특징으로 하는 사전행동적 사용자 인터페이스.

**청구항 13.**

제1항 내지 제12항 중 어느 한항에 있어서,

상기 사용자 및/또는 상기 운영체제와의 상호작용들의 결과로서 상기 학습모듈에 의해 수집된 정보를 보유하기 위한 지식베이스를 추가로 포함하는 것을 특징으로 하는 사전행동적 사용자 인터페이스.

**청구항 14.**

제13항에 있어서,

상기 학습모듈은 상기 운영체제의 상태를 인지하기 위한 복수의 센서들을 포함하는 것을 특징으로 하는 사전행동적 사용자 인터페이스.

**청구항 15.**

제14항에 있어서,

상기 학습모듈은 상기 센서들로부터의 출력을 인지하여 상기 사용자 인터페이스의 상태와 상기 운영체제의 상태를 결정하는 인지부를 추가로 포함하는 것을 특징으로 하는 사전행동적 사용자 인터페이스.

**청구항 16.**

제15항에 있어서,

상기 학습모듈은 상기 지식베이스를 생성하고 상기 운영체제의 상태와 상기 사용자 인터페이스의 변경 사이의 상호 관계를 학습하는 추론 시스템을 추가로 포함하는 것을 특징으로 하는 사전행동적 사용자 인터페이스.

**청구항 17.**

제13항에 있어서,

상기 사용자로의 행동으로부터 및 미리 결정된 정보로부터 결정된 복수의 통합된 지식 베이스들을 추가로 포함하는 것을 특징으로 하는 사전행동적 사용자 인터페이스.

**청구항 18.**

제1항 내지 제17항 중 어느 한항에 있어서,

상기 학습모듈은 인공 지능 알고리즘 및 기계 학습 알고리즘 중 적어도 하나를 추가로 포함하는 것을 특징으로 하는 사전행동적 사용자 인터페이스.

**청구항 19.**

제1항 내지 제18항 중 어느 한항에 있어서,

상기 사용자의 행동을 모델화하는 사용자 모델을 추가로 포함하는 것을 특징으로 하는 사전행동적 사용자 인터페이스.

**청구항 20.**

제19항에 있어서,

상기 학습모듈은 상기 사용자 인터페이스에 대한 사용자의 반응에 따라 상기 사용자의 비명시적인 성향(implicit preference)을 검출하기 위해 상기 사용자 모델을 사용하는 것을 특징으로 하는 사전행동적 사용자 인터페이스.

**청구항 21.**

사용자 인터페이스를 통해 사용자와 운영체제를 갖춘 연산장치와의 사이에서 사전행동적 상호작용을 하는 방법에 있어서,

상기 사용자 인터페이스와 상기 사용자의 적어도 하나의 상호작용에 따라 사용자 행동의 적어도 하나의 패턴을 검출하는 단계와;

상기 패턴에 따라 상기 사용자 인터페이스의 적어도 하나의 기능을 사전행동적으로 변경하는 단계를 포함하는 사전행동적 상호작용 방법.

**청구항 22.**

제21항에 있어서,

상기 적어도 하나의 패턴은 상기 사용자 인터페이스와 상기 사용자의 적어도 하나의 이전 상호작용에 따라 결정된 패턴과, 미리 결정된 패턴, 또는 그 조합으로 이루어진 군으로부터 선택되는 것을 특징으로 하는 사전행동적 상호작용 방법.

### 청구항 23.

제21항 또는 제22항에 있어서,

상기 사용자 인터페이스는 그래픽 표시부를 특징적 요소로서 가지며,

상기 사용자 인터페이스의 적어도 하나의 기능을 변경하는 단계는 상기 그래픽 표시부의 적어도 일부분을 변경하는 단계를 포함하는 것을 특징으로 하는 사전행동적 상호작용 방법.

### 청구항 24.

제23항에 있어서,

상기 그래픽 표시부의 적어도 일부분을 변경하는 단계는

상기 검출된 패턴에 따라 표시할 메뉴를 선택하는 단계와;

상기 메뉴를 표시하는 단계를 포함하는 것을 특징으로 하는 사전행동적 상호작용 방법.

### 청구항 25.

제24항에 있어서,

상기 메뉴를 선택하는 단계는 복수의 메뉴 옵션으로부터 하나의 메뉴를 구성하는 단계를 포함하는 것을 특징으로 하는 사전행동적 상호작용 방법.

### 청구항 26.

제21항 또는 제22항에 있어서,

상기 사용자 인터페이스는 오디오 표시부를 특징적 요소로서 포함하며, 상기 사용자 인터페이스의 적어도 하나의 기능을 변경하는 단계는 상기 연산장치에 의해 생성된 적어도 하나의 가청 사운드를 변경하는 단계를 포함하는 것을 특징으로 하는 사전행동적 상호작용 방법.

### 청구항 27.

제21항 내지 제26항 중 어느 한항에 있어서,

상기 연산장치는 개인용 컴퓨터와, ATM과, 이동 정보 장치, 혹은 운영체제를 갖춘 소비재로 이루어진 군으로부터 선택되는 것을 특징으로 하는 사전행동적 상호작용 방법.

### 청구항 28.

제21항 내지 제27항 중 어느 한항에 있어서,

상기 학습 모듈은 사전행동적 변경으로부터 직접적인 사용자의 선택으로 이어지는 상기 사전행동적 변경의 확률을 최대화하는 것을 특징으로 하는 사전행동적 상호작용 방법.

### 청구항 29.

제28항에 있어서,

상기 최대화는 학습 보강을 통해 수행되는 것을 특징으로 하는 사전행동적 상호작용 방법.

### 청구항 30.

제29항에 있어서,

상기 학습보강은 반복적 학습 과정을 통해 수행되는 것을 특징으로 하는 사전행동적 상호작용 방법.

**청구항 31.**

제30항에 있어서,

상기 학습 과정의 반복(iteration)은 각각 상기 변경이 수행된 후에 행해지는 것을 특징으로 하는 사전행동적 상호작용 방법.

**청구항 32.**

제21항 내지 제31항 중 어느 한항에 있어서,

상기 사용자 인터페이스의 적어도 하나의 기능을 사전행동적으로 변경하는 단계는 상기 운영체제를 통해 부가 소프트웨어 애플리케이션을 작동시키는 단계를 포함하는 것을 특징으로 하는 사전행동적 상호작용 방법.

**청구항 33.**

제21항 내지 제32항 중 어느 한항에 있어서,

상기 사용자 및/또는 상기 운영체제와의 상호작용들의 결과로서 상기 학습모듈에 의해 수집된 정보를 보유하기 위한 지식베이스를 추가로 포함하는 것을 특징으로 하는 사전행동적 상호작용 방법.

**청구항 34.**

제33항에 있어서,

상기 학습모듈은 상기 운영체제의 상태를 인지하기 위한 복수의 센서들을 포함하는 것을 특징으로 하는 사전행동적 상호작용 방법.

**청구항 35.**

제34항에 있어서,

상기 학습모듈은 상기 센서들로부터의 출력을 인지하여 상기 사용자 인터페이스의 상태와 상기 운영체제의 상태를 결정하는 인지부를 추가로 포함하는 것을 특징으로 하는 사전행동적 상호작용 방법.

**청구항 36.**

제35항에 있어서,

상기 학습모듈은 상기 지식베이스를 개신하고 상기 운영체제의 상태와 상기 사용자 인터페이스의 변경 사이의 상호관계를 학습하는 추론 시스템을 추가로 포함하는 것을 특징으로 하는 사전행동적 상호작용 방법.

**청구항 37.**

제33항에 있어서,

상기 사용자로의 행동으로부터 및 미리 결정된 정보로부터 결정된 복수의 통합된 지식 베이스들을 추가로 포함하는 것을 특징으로 하는 사전행동적 상호작용 방법.

**청구항 38.**

제21항 내지 제37항 중 어느 한항에 있어서,

상기 학습모듈은 인공 지능 알고리즘 및 기계 학습 알고리즘 중 적어도 하나를 추가로 포함하는 것을 특징으로 하는 사전행동적 상호작용 방법.

**청구항 39.**

제21항 내지 제38항 중 어느 한항에 있어서,

상기 사용자의 행동을 모델화하는 사용자 모델을 추가로 포함하는 것을 특징으로 하는 사전행동적 상호작용 방법.

**청구항 40.**

제39항에 있어서.

상기 학습모듈은 상기 사용자 인터페이스에 대한 사용자의 반응에 따라 상기 사용자의 비명시적인 성향을 검출하기 위해 상기 사용자 모델을 사용하는 것을 특징으로 하는 사전행동적 상호작용 방법.

#### 청구항 41.

운영체제를 가지며, 사용자와 상호작용을 하기 위한 사전행동적 연산장치에 있어서,

a) 상기 사용자와 상기 운영체제 사이의 통신을 위한 사용자 인터페이스;

b) 상기 사용자 인터페이스와 상기 사용자의 상호작용의 적어도 하나의 패턴을 검출하고, 상기 검출된 패턴에 따라 상기 사용자 인터페이스의 적어도 하나의 기능을 사전행동적으로 변경하는 학습모듈을 포함하여 이루어지는 사전 행동적 연산장치.

#### 청구항 42.

운영체제를 갖는 이동 정보 장치를 위한 행동형 시스템(behavioral system)에 있어서,

상기 이동 정보 장치를 위한 적응형 시스템을 포함하며,

상기 적응형 시스템은 상기 이동 정보 장치의 적어도 하나의 기능을 변경하기 위해 직접적인 사용자의 명령보다는 사용자 행동의 분석에 따라 상기 이동 정보 장치의 적어도 하나의 기능을 변경하는 것을 특징으로 하는 행동형 시스템(behavioral system).

#### 청구항 43.

제42항에 있어서.

상기 운영체제는 내장형 시스템을 포함하는 것을 특징으로 하는 행동형 시스템.

#### 청구항 44.

제42항 또는 제43항에 있어서.

상기 이동 정보 장치는 휴대폰을 포함하는 것을 특징으로 하는 행동형 시스템.

#### 청구항 45.

제40항 내지 제44항 중 어느 한항에 있어서,

상기 사용자 행동의 상기 분석은 상기 이동 정보 장치와 복수의 사용자의 상호작용들의 분석을 포함하는 것을 특징으로 하는 행동형 시스템.

#### 청구항 46.

제45항에 있어서,

상기 분석은 적어도 하나의 미리 결정된 패턴과 상기 복수의 사용자의 상호작용들을 비교하는 것을 추가로 포함하며, 상기 적어도 하나의 미리 결정된 패턴은 상기 적어도 하나의 기능과 연관되는 것을 특징으로 하는 행동형 시스템.

#### 청구항 47.

제45항 또는 제46항에 있어서,

상기 분석은 이미 검출된 사용자 행동의 적어도 하나의 패턴과 상기 복수의 사용자의 상호작용들을 비교하는 것을 추가로 포함하고, 상기 이미 검출된 사용자 행동의 상기 적어도 하나의 패턴은 상기 적어도 하나의 기능을 변경하는 것과 연관되는 것을 특징으로 하는 행동형 시스템.

#### 청구항 48.

제42항 내지 제47항 중 어느 한항에 있어서,

상기 적어도 하나의 기능은 상기 이동 정보 장치에 의한 오디오 기능(audio capability)을 포함하는 것을 특징으로 하는 행동형 시스템.

**청구항 49.**

제48항에 있어서,

상기 오디오 기능은 상기 이동 정보 장치에 의한 가청 사운드를 생성하는 것을 포함하는 것을 특징으로 하는 행동형 시스템.

**청구항 50.**

제49항에 있어서,

상기 가청 사운드는 벨소리, 알람소리 및 착신메시지소리 중 적어도 하나를 포함하는 것을 특징으로 하는 행동형 시스템.

**청구항 51.**

제48항에 있어서,

상기 오디오 기능은 상기 이동 정보 장치에 의한 가청 사운드의 수신 및 번역(interpret)을 포함하는 것을 특징으로 하는 행동형 시스템.

**청구항 52.**

제42항 내지 제47항 중 어느 한항에 있어서,

상기 적어도 하나의 기능은 상기 이동 정보 장치에 의한 화상 표시와 관련되는 것을 특징으로 하는 행동형 시스템.

**청구항 53.**

제52항에 있어서,

상기 적어도 하나의 기능은 메뉴의 표시를 포함하는 것을 특징으로 하는 행동형 시스템.

**청구항 54.**

제42항 내지 제53항 중 어느 한항에 있어서,

상기 적응형 시스템은 상기 이동 정보 장치에 의해 동작되는 것을 특징으로 하는 행동형 시스템.

**청구항 55.**

제42항 내지 제53항 중 어느 한항에 있어서,

상기 이동 정보 장치는 네트워크를 통해 통신이 가능한 것을 특징으로 하는 행동형 시스템.

**청구항 56.**

제55항에 있어서,

상기 적응형 시스템은 적어도 부분적으로 상기 네트워크로부터 상기 이동 정보 장치로 전송된 명령에 따라 동작되는 것을 특징으로 하는 행동형 시스템.

**청구항 57.**

제56항에 있어서,

상기 적응형 시스템의 적어도 하나의 동작과 관련되는 데이터는 상기 이동 정보 장치 이외의 위치에 저장되며, 상기 위치는 상기 네트워크를 통해 접근이 가능한 것을 특징으로 하는 행동형 시스템.

**청구항 58.**

제42항 내지 제57항 중 어느 한항에 있어서,

학습모듈을 추가로 포함하며, 상기 분석은 상기 학습모듈에 의해 수행되고, 상기 학습모듈은 인공 지능(AI) 알고리즘, 기계 학습(ML) 알고리즘 또는 유전적 알고리즘 중 적어도 하나에 따라 상기 사용자 행동에 적응되는 것을 특징으로 하는 행동형 시스템.

**청구항 59.**

사전행동적 이동 정보 장치에 있어서,

상기 이동 정보 장치를 위한 적응형 시스템을 포함하며,

상기 적응형 시스템은 상기 이동 정보 장치의 적어도 하나의 기능을 변경하기 위해 직접적인 사용자의 명령보다는 사용자 행동의 분석에 따라 상기 이동 정보 장치의 적어도 하나의 기능을 변경하는 것을 특징으로 하는 사전행동적 이동 정보 장치.

**청구항 60.**

복수의 사용자의 이동 정보 장치와의 상호작용들을 분석하여, 분석결과 형성하는 단계와;

상기 이동 정보 장치의 적어도 하나의 기능을 변경하기 위해 직접적인 사용자의 명령보다는 상기 분석결과에 따라 상기 이동 정보 장치의 적어도 하나의 기능을 사전행동적으로 변경하는 단계를 포함하는 사용자를 위한 이동 정보 장치의 적어도 하나의 기능에 적응하기 위한 방법.

**청구항 61.**

이동 정보 장치 망을 통해서 이동 정보 장치와 함께 사용하기 위한 지능형 에이전트에 있어서,

상기 지능형 에이전트와의 사용자 인터페이스를 제공하는 아바타와; 상기 이동 정보 장치 망을 통해 상기 이동 정보 장치의 상호작용을 제어하는 에이전트를 포함하는 지능형 에이전트.

**청구항 62.**

제61항에 있어서,

상기 아바타 및 상기 에이전트는 상기 이동 정보 장치에 의해 동작되는 것을 특징으로 하는 지능형 에이전트.

**청구항 63.**

제61항 또는 제62항에 있어서,

상기 이동 정보 장치는 적어도 하나의 다른 이동 정보 장치와 통신하며, 상기 적어도 하나의 이동 정보 장치는 제2 에이전트와 연관되어 있어, 상기 에이전트는 상기 제2 에이전트와 통신이 가능한 것을 특징으로 하는 지능형 에이전트.

**청구항 64.**

제63항에 있어서,

상기 이동 정보 장치는 상기 이동 정보 장치 망을 통해 적어도 하나의 다른 이동 정보 장치와 통신하는 것을 특징으로 하는 지능형 에이전트.

**청구항 65.**

제63항에 있어서,

상기 이동 정보 장치는 상기 이동 정보 장치 망이 없이 적어도 하나의 다른 이동 정보 장치와 직접 통신하는 것을 특징으로 하는 지능형 에이전트.

**청구항 66.**

제63항 내지 제65항 중 어느 한항에 있어서,

상기 적어도 하나의 다른 이동 정보 장치는 제2 아바타와 연관되어 있고, 상기 이동 정보 장치의 사용자와 상기 적어도 다른 이동 정보 장치의 사용자가 상기 아바타 및 상기 제2 아바타를 통해 통신하는 것을 특징으로 하는 지능형 에이전트.

**청구항 67.**

제65항에 있어서,

상기 통신은 게임에 관련된 것을 특징으로 하는 지능형 에이전트.

**청구항 68.**

제67항에 있어서,

상기 게임은 롤플레잉(role-playing) 게임을 포함하는 것을 특징으로 하는 지능형 에이전트.

**청구항 69.**

제61항 내지 제68항 중 어느 한항에 있어서,

상기 아바타 또는 상기 에이전트 중 적어도 하나는 적어도 부분적으로 상기 이동 정보 장치 망에서 상기 이동 정보 장치로 전송된 명령에 따라 동작되는 것을 특징으로 하는 지능형 에이전트.

**청구항 70.**

제69항에 있어서,

상기 아바타 또는 상기 에이전트 중 적어도 하나의 적어도 하나의 동작과 연관된 데이터가 상기 이동 정보 장치 이외의 위치에 저장되고, 상기 위치는 상기 이동 정보 장치 망을 통해 접근이 가능한 것을 특징으로 하는 지능형 에이전트.

**청구항 71.**

제61항 내지 제70항 중 어느 한항에 있어서,

상기 아바타의 모습의 적어도 하나의 특성이 변경 가능한 것을 특징으로 하는 지능형 에이전트.

**청구항 72.**

제71항에 있어서,

상기 아바타의 모습의 적어도 하나의 특성이 사용자 명령에 따라 변경 가능한 것을 특징으로 하는 지능형 에이전트.

**청구항 73.**

제71항에 있어서,

상기 아바타의 모습의 적어도 하나의 특성이 미리 결정된 아바타 피부에 따라 변경이 가능한 것을 특징으로 하는 지능형 에이전트.

**청구항 74.**

제73항에 있어서,

상기 미리 결정된 아바타 피부는 상기 사용자에 의해 미리 결정되는 것을 특징으로 하는 지능형 에이전트.

**청구항 75.**

제71항 내지 제74항 중 어느 한항에 있어서,

상기 아바타의 모습의 적어도 하나의 특성이 자동화된 진화 알고리즘에 따라 변경이 가능한 것을 특징으로 하는 지능형 에이전트.

**청구항 76.**

제75항에 있어서,

상기 자동화된 진화 알고리즘은 유전적 알고리즘을 포함하는 것을 특징으로 하는 지능형 에이전트.

**청구항 77.**

제61항 내지 제76항 중 어느 한 항에 있어서,

상기 이동 정보 장치 망은 상기 이동 정보 장치의 물리적 위치를 결정하기 위한 위치 결정부를 포함하고, 상기 사용자는 상기 에이전트의 행위를 통해 상기 물리적 위치에 대한 정보를 요청할 수 있는 것을 특징으로 하는 지능형 에이전트.

**청구항 78.**

제77항에 있어서,

상기 위치 결정부는 상기 이동 정보 장치의 상기 물리적 위

치에 대해 상대적인 제2 물리적 위치를 결정할 수 있고, 상기 사용자는 상기 에이전트의 행위를 통해 상기 제2 물리적 위치에 대한 정보를 요청할 수 있는 것을 특징으로 하는 지능형 에이전트.

**청구항 79.**

제78항에 있어서,

상기 사용자는 카테고리에 따라 상기 제2 물리적 위치를 요청하는 것을 특징으로 하는 지능형 에이전트.

**청구항 80.**

제79항에 있어서,

상기 카테고리는 상업적 위치, 의료적 위치, 및 공공 안전 위치로 이루어진 군으로부터 선택되는 것을 특징으로 하는 지능형 에이전트.

**청구항 81.**

제79항에 있어서,

상기 카테고리는 상업적 위치를 포함하고, 상기 상업적 위치는 상기 에이전트의 상기 행위에 따라 상기 이동 정보 장치로 메시지를 전송하는 것을 특징으로 하는 지능형 에이전트.

**청구항 82.**

제81항에 있어서,

상기 메시지는 광고 또는 쿠폰, 또는 그 조합 중 적어도 하나를 포함하는 것을 특징으로 하는 지능형 에이전트.

**청구항 83.**

제81항 또는 제82항에 있어서,

상기 에이전트는 적어도 하나의 판단기준에 따라 상기 메시지를 필터링하는 것을 특징으로 하는 지능형 에이전트.

**청구항 84.**

제81항 내지 제83항 중 어느 한항에 있어서,

상기 아바타는 적어도 상기 메시지에 대한 정보를 상기 사용자에게 제시하는 것을 특징으로 하는 지능형 에이전트.

**청구항 85.**

제78항 내지 제84항 중 어느 한항에 있어서,

상기 사용자는 상기 아바타를 통해 상기 제2 물리적 위치에 대한 정보를 요청하는 것을 특징으로 하는 지능형 에이전트.

**청구항 86.**

제61항 내지 제85항 중 어느 한항에 있어서,

상기 이동 정보 장치 망은 가상의 상업용 위치와 통신하며, 상기 사용자는 상기 아바타를 통해 상기 가상의 상업적 위치와 통신하는 것을 특징으로 하는 지능형 에이전트.

**청구항 87.**

제86항에 있어서,

상기 사용자는 상기 아바타를 통해 상기 가상의 상업적 위치와 구입 행위를 하는 것을 특징으로 하는 지능형 에이전트.

**청구항 88.**

제86항 또는 제87항에 있어서,

상기 사용자는 상기 에이전트를 통해 상기 가상의 상업적 위치를 탐색하는 것을 특징으로 하는 지능형 에이전트.

**청구항 89.**

제86항 내지 제88항 중 어느 한항에 있어서,

상기 아바타는 상기 가상의 상업적 위치로부터 구입된 액세서리를 전달받을 수 있는 것을 특징으로 하는 지능형 에이전트.

**청구항 90.**

제61항 내지 89항 중 어느 한항에 있어서,

상기 이동 정보 장치는 소프트웨어를 수신할 수 있고, 상기 에이전트는 상기 이동 정보 장치 상으로의 상기 소프트웨어의 설치 중 적어도 일부분을 수행하는 것을 특징으로 하는 지능형 에이전트.

**청구항 91.**

제90항에 있어서,

상기 사용자는 상기 소프트웨어의 환경설정(configuration) 중 적어도 일부분을 수행하기 위하여 상기 아바타와 상호작용하는 것을 특징으로 하는 지능형 에이전트.

**청구항 92.**

제61항 내지 제91항 중 어느 한항에 있어서,

상기 에이전트는 상기 사용자를 교육시키기 위한 교육모듈을 추가로 포함하는 것을 특징으로 하는 지능형 에이전트.

**청구항 93.**

제92항에 있어서,

상기 교육모듈은 상기 이동 정보 장치의 적어도 하나의 요소(aspect)에 대해 상기 사용자에게 교육하도록 동작 가능한 것을 특징으로 하는 지능형 에이전트.

**청구항 94.**

제92항에 있어서,

상기 교육모듈은 상기 이동 정보 장치 외부의 적어도 하나의 대상에 대해 상기 사용자에게 교육하도록 동작 가능한 것을 특징으로 하는 지능형 에이전트.

**청구항 95.**

제61항 내지 제94항 중 어느 한항에 있어서,

상기 아바타는 상기 사용자에게 오락을 제공할 수 있는 것을 특징으로 하는 지능형 에이전트.

**청구항 96.**

사용자를 위한 생존형 이동 전화(living mobile telephone)에 있어서,

상기 이동 전화의 상기 사용자와의 상호작용을 제어하기 위한 에이전트와;

상기 에이전트와의 사용자 인터페이스를 제공하기 위한 아바타를 포함하는 것을 특징으로 하는 생존형 이동 전화.

**청구항 97.**

제96항에 있어서,

상기 이동 전화는 이동 전화망에 연결되고, 상기 에이전트는 상기 이동전화의 상기 이동 전화망과의 상호작용을 제어하는 것을 특징으로 하는 생존형 이동 전화.

**청구항 98.**

제95항 또는 제96항에 있어서,

상기 에이전트는

(i) 인공지능(AI) 시스템과;

(ii) 지식베이스를 포함하여 이루어지며,

상기 인공지능 시스템은 적어도 하나의 행위를 결정하기 위하여 상기 지식베이스로부터 정보를 획득하는 것을 특징으로 하는 생존형 이동 전화.

**청구항 99.**

제98항에 있어서,

상기 지시베이스는 적어도 하나의 사용자 성향(user preference)에 대한 정보를 포함하는 것을 특징으로 하는 생존형 이동 전화.

**청구항 100.**

제98항 또는 제99항에 있어서,

상기 지식베이스는 적어도 하나의 사용자 행위 패턴에 대한 정보를 포함하는 것을 특징으로 하는 생존형 이동 전화.

**청구항 101.**

제98항 내지 제100항 중 어느 한항에 있어서,

상기 지식베이스는 상기 에이전트에 의한 행위를 위한 상기 사용자에 대한 정보를 포함하는 것을 특징으로 하는 생존형 이동 전화.

**청구항 102.**

제101항에 있어서,

상기 지식베이스는 상기 사용자에 의한 거절 이후에 행위를 반복하는 것에 대한 상기 에이전트를 위한 행동억제요인(disincentive)을 포함하는 것을 특징으로 하는 생존형 이동 전화.

**청구항 103.**

제98항 내지 제102항 중 어느 한항에 있어서,

상기 인공지능 시스템은 행위를 수행하는 것에 대한 상기 에이전트를 위한 보상(reward)을 결정하여, 상기 보상에 따라 상기 행위가 선택되는 것을 특징으로 하는 생존형 이동 전화.

#### 청구항 104.

제103항에 있어서,

상기 보상은 상기 지식베이스 안의 정보에 따라 결정되는 것을 특징으로 하는 생존형 이동 전화.

#### 청구항 105.

제103항 또는 제104항에 있어서,

상기 인공지능 시스템은 복수의 행위들을 모의(simulate)하고, 상기 보상에 따라 상기 에이전트를 위한 행위를 선택하는 것을 특징으로 하는 생존형 이동 전화.

#### 청구항 106.

제98항 내지 제105항 중 어느 한항에 있어서,

상기 인공지능 시스템은 사용자 인터페이스의 적어도 일부분을 결정하는 것을 특징으로 하는 생존형 이동 전화.

#### 청구항 107.

제106항에 있어서,

상기 사용자 인터페이스는 적어도 하나의 메뉴를 포함하는 것을 특징으로 하는 생존형 이동 전화.

#### 청구항 108.

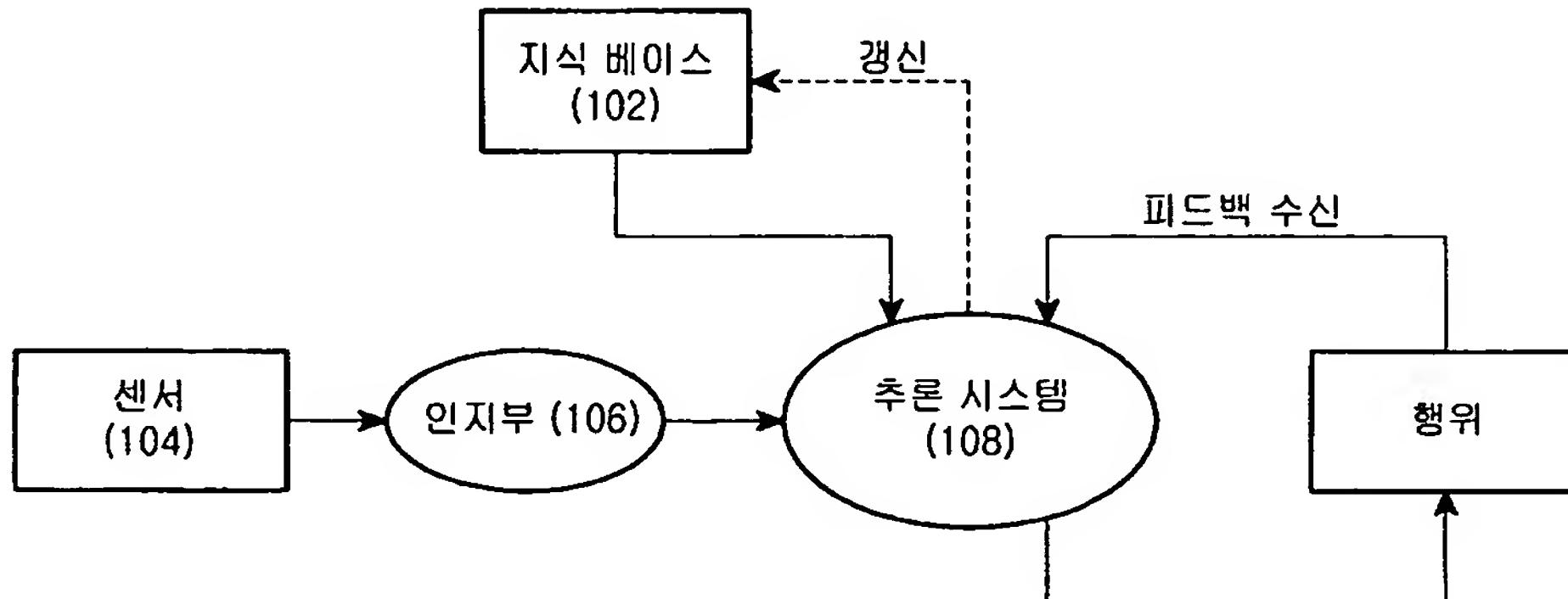
제107항에 있어서,

상기 인공지능 시스템은 상기 지식베이스 내의 정보에 따라 상기 적어도 하나의 메뉴를 부분적으로 구성하는 것을 특징으로 하는 생존형 이동 전화.

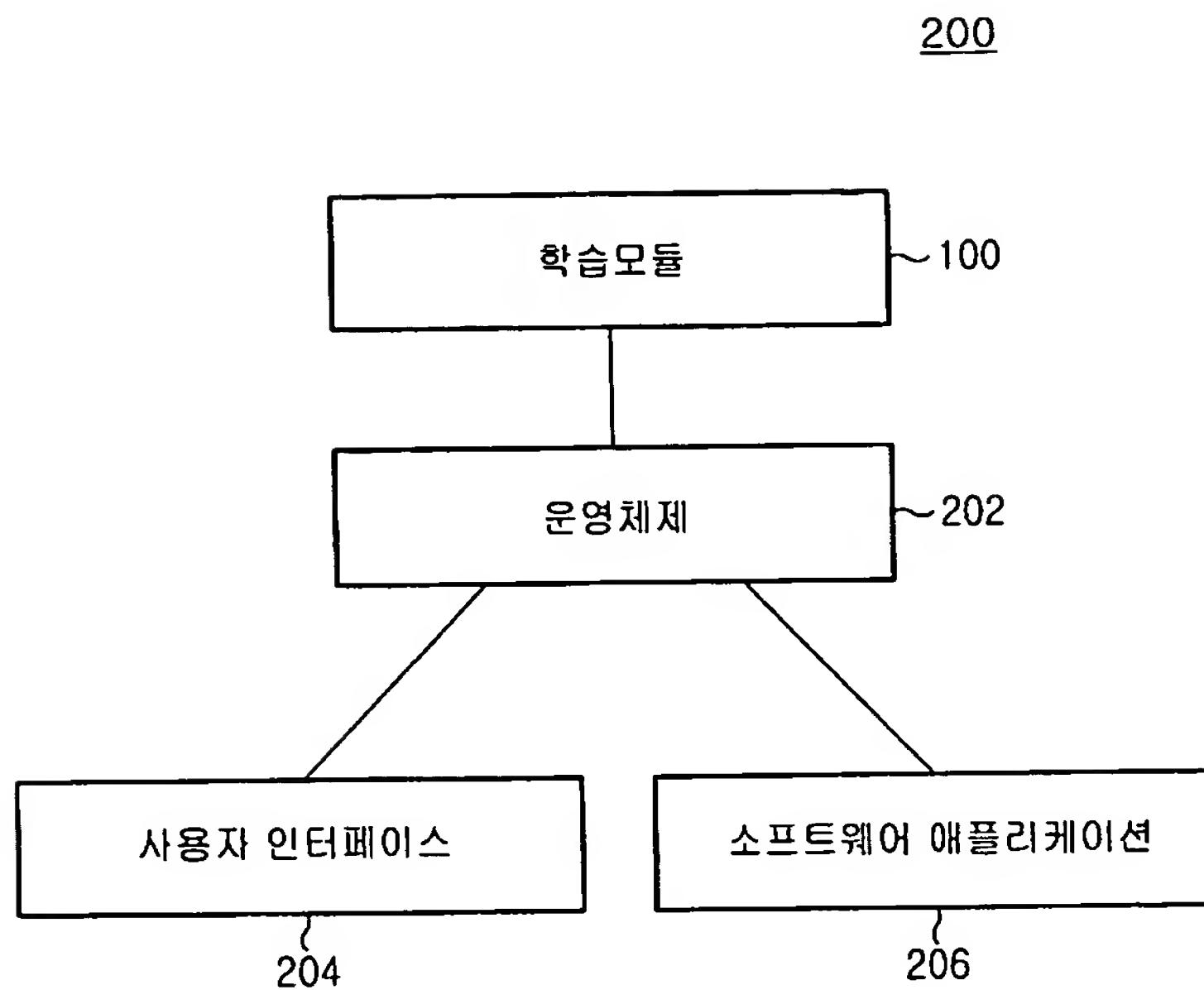
도면

도면 1

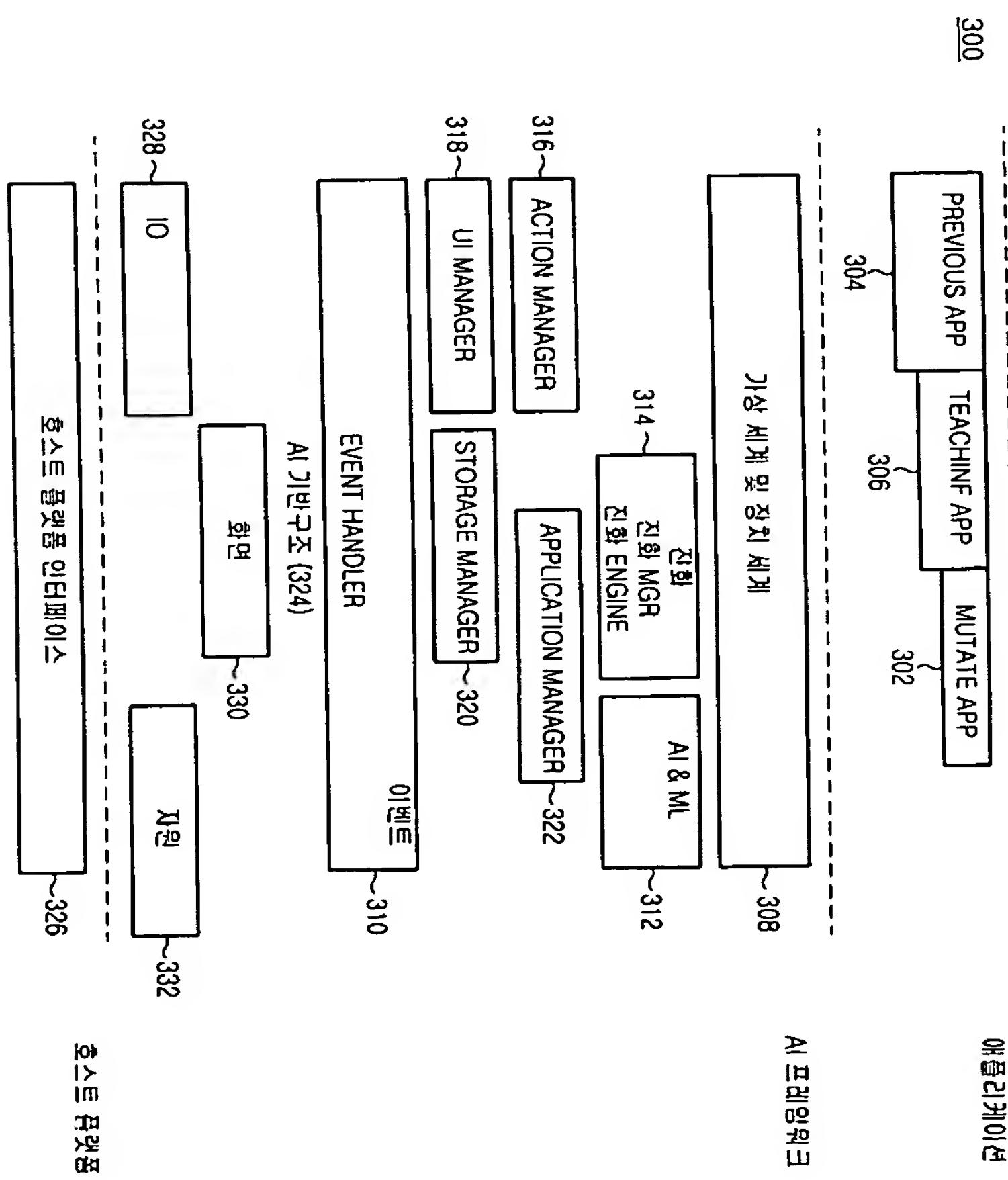
100



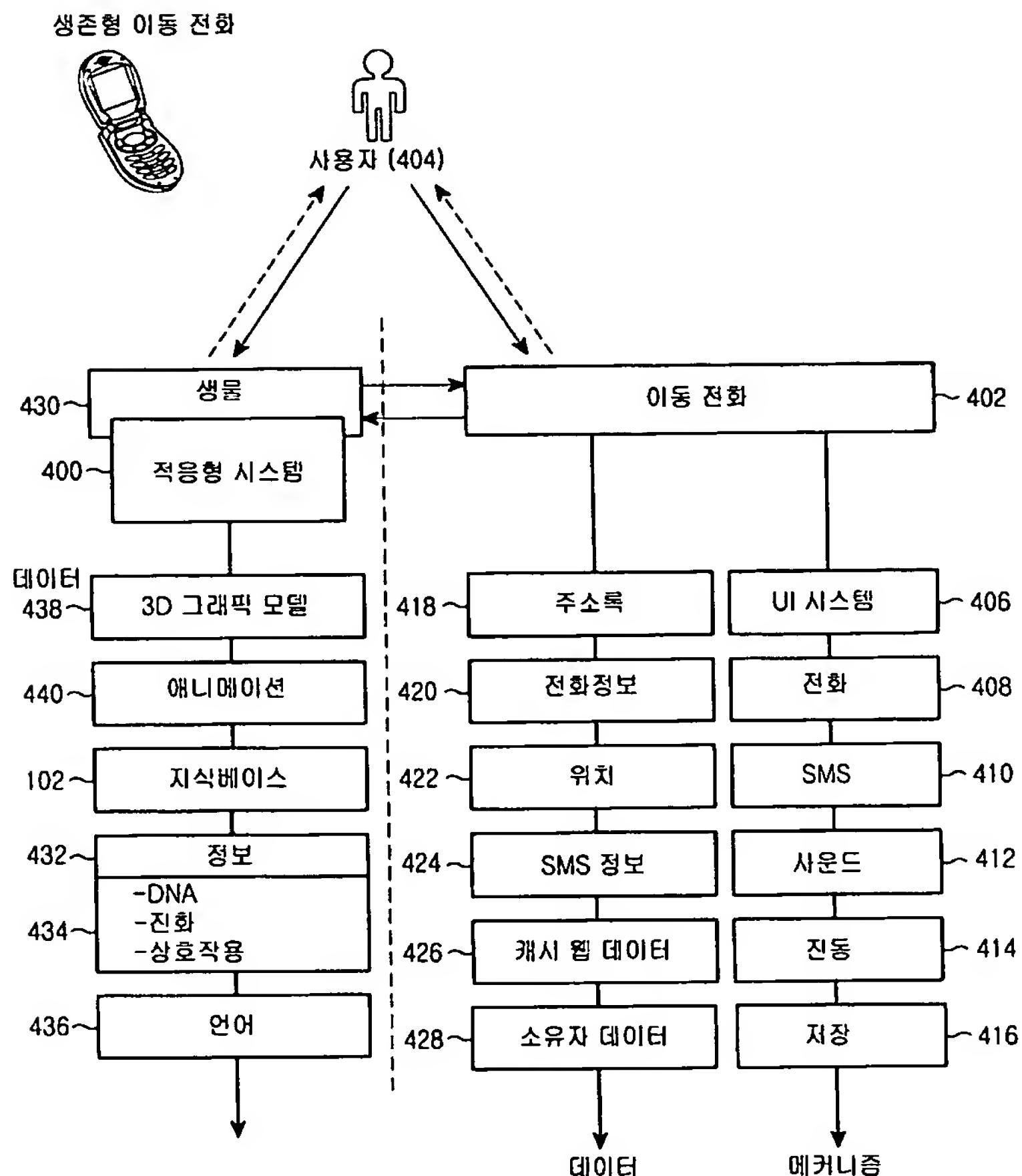
도면2



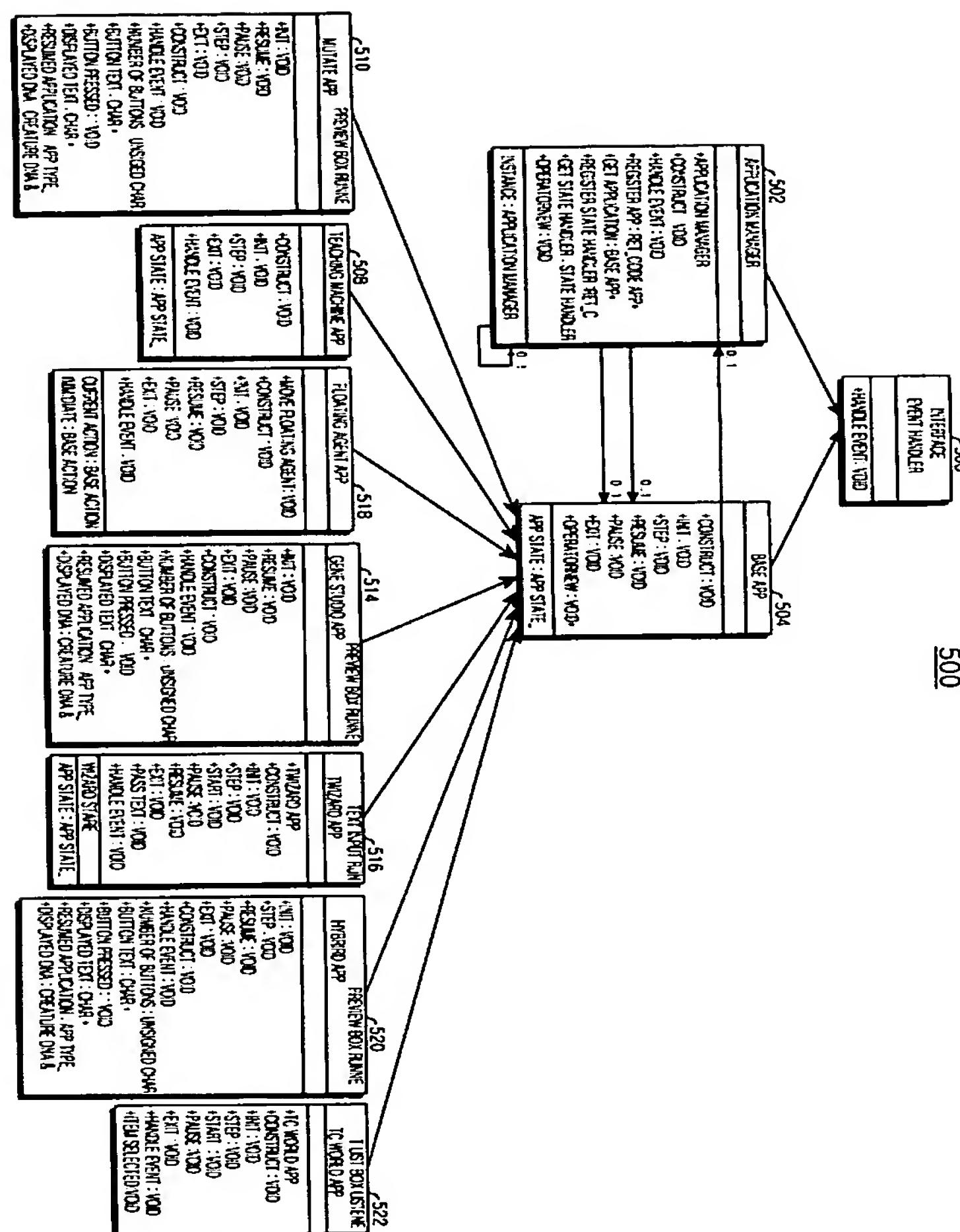
도면3



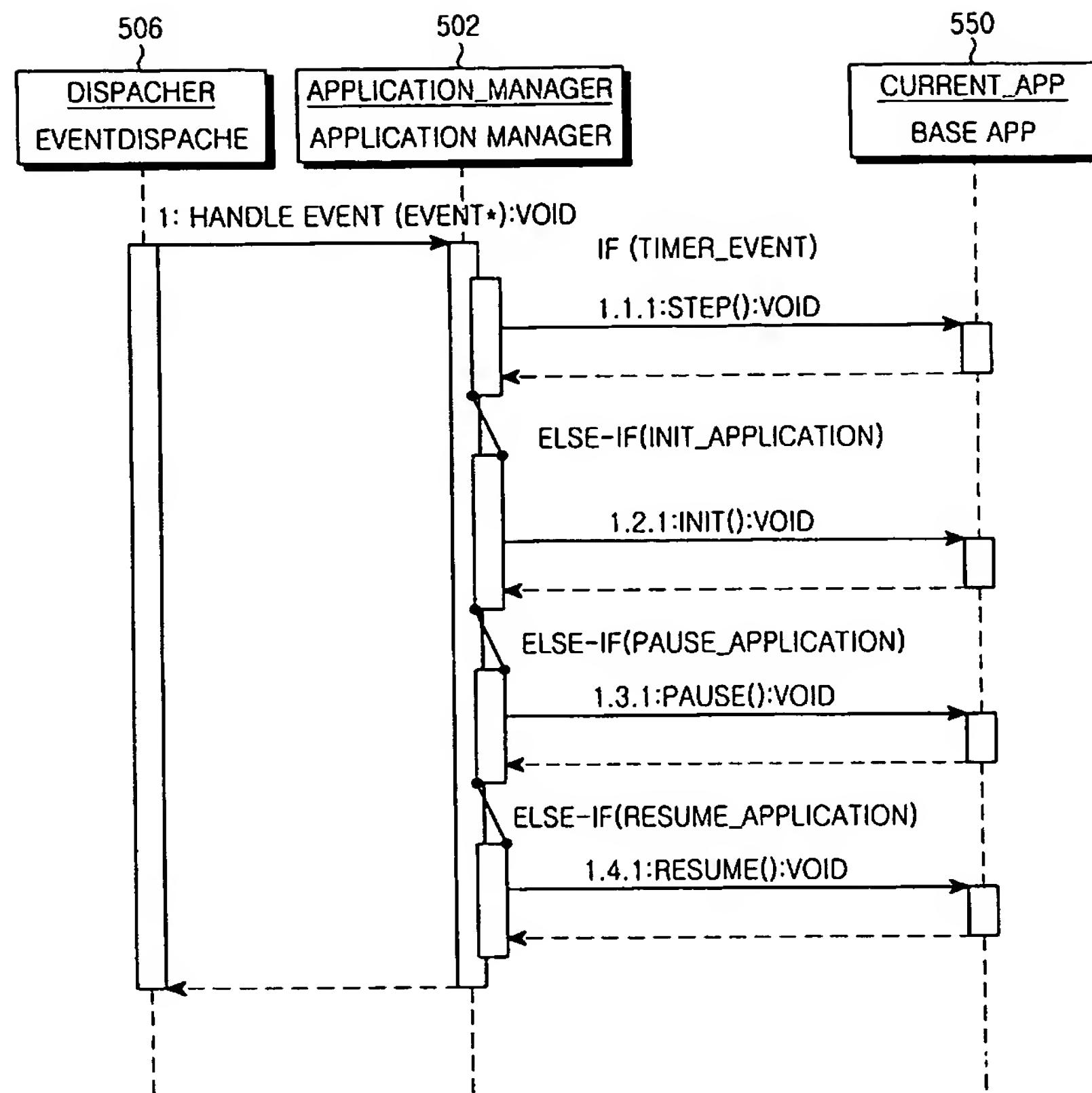
도면4



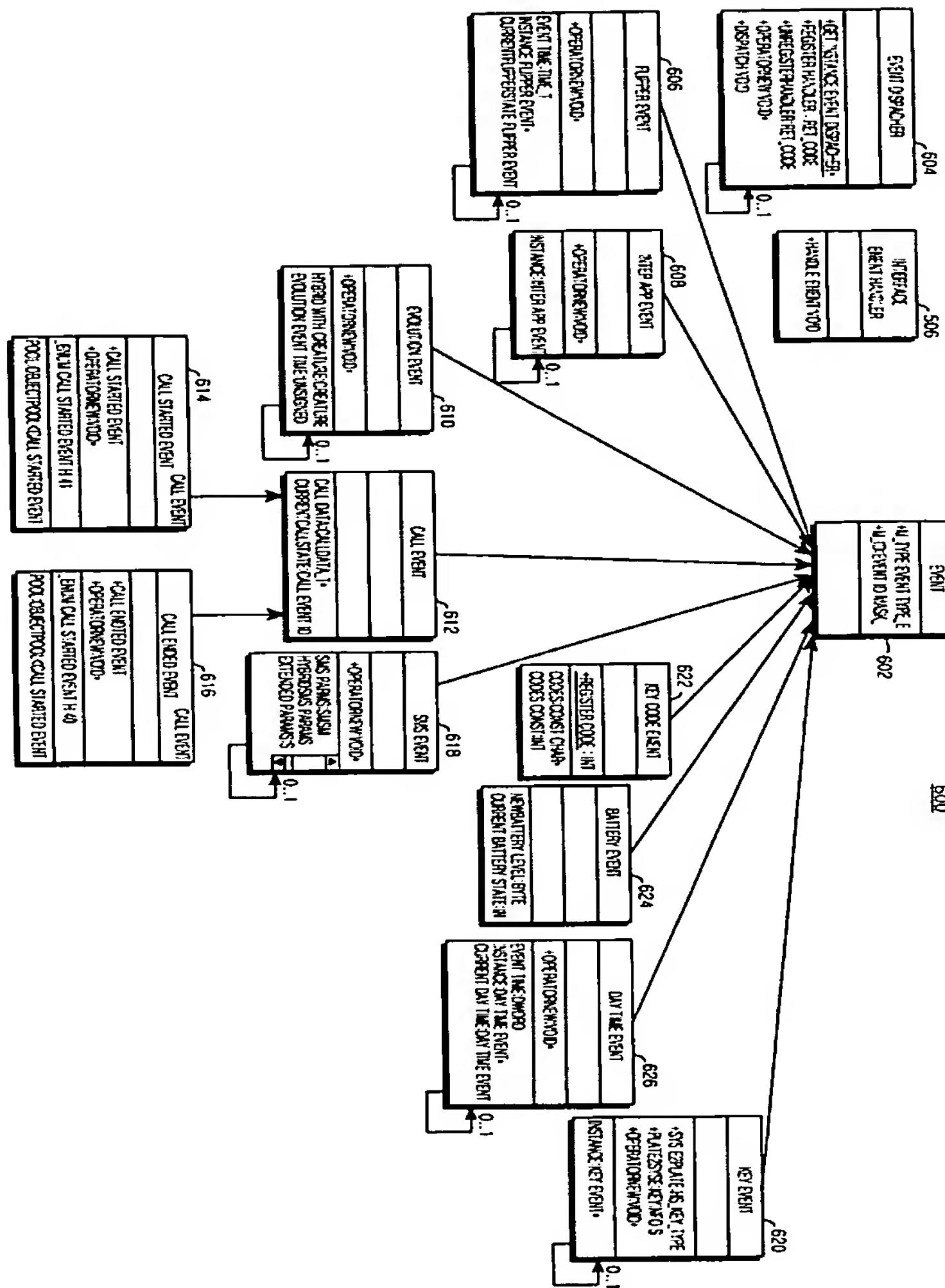
### 도면5a



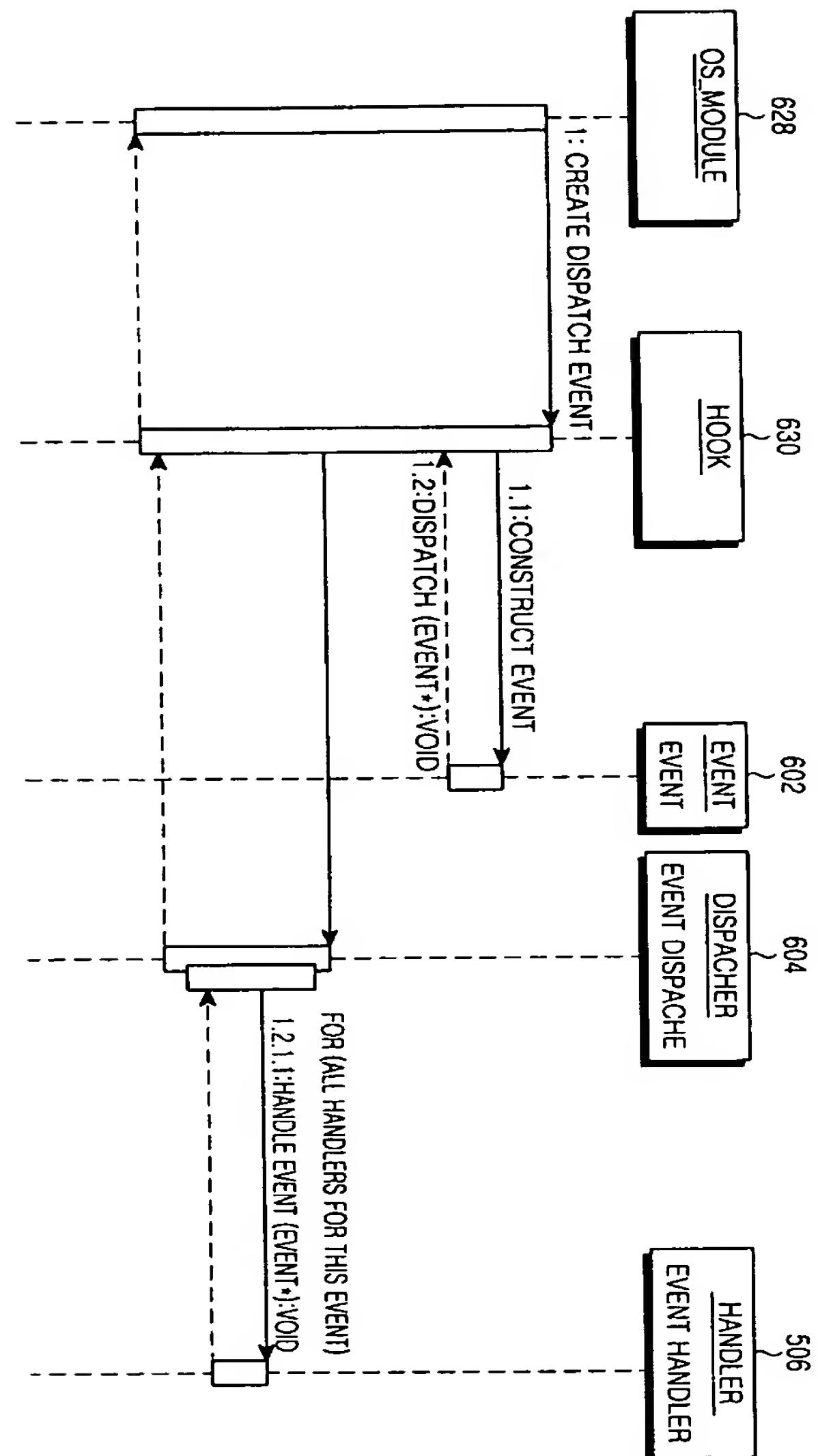
도면5b



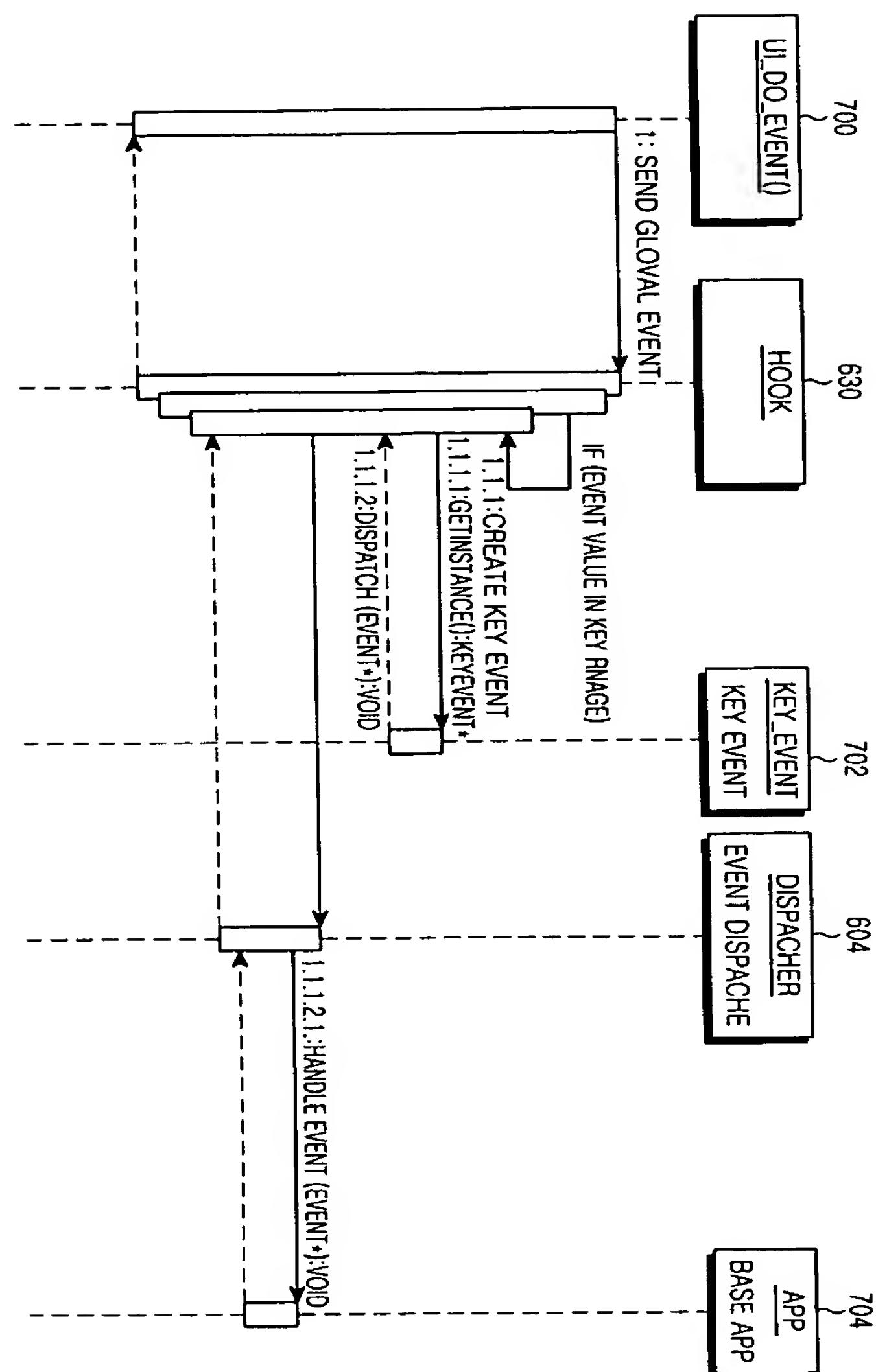
### 도면6a



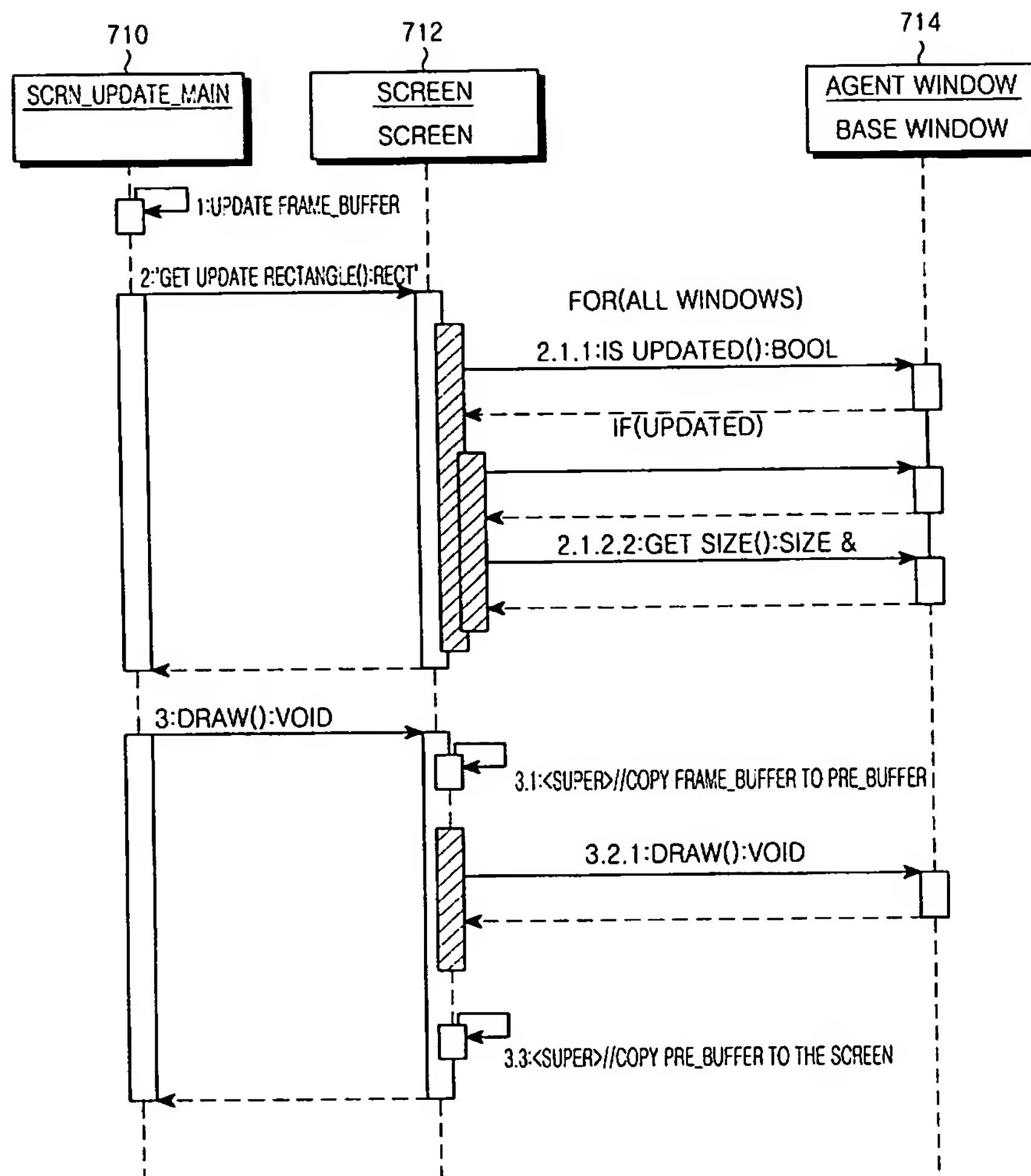
도면6b



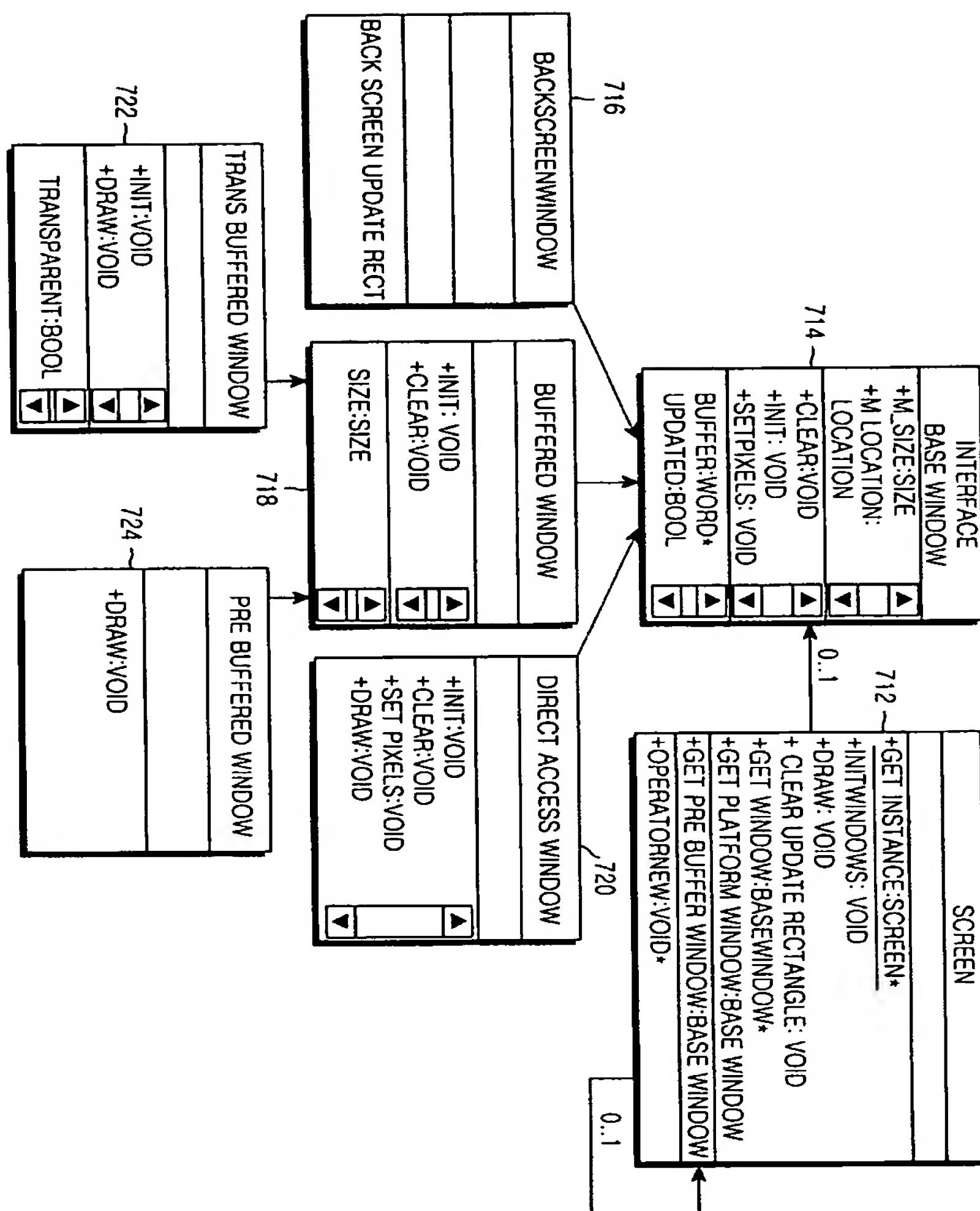
도면7a



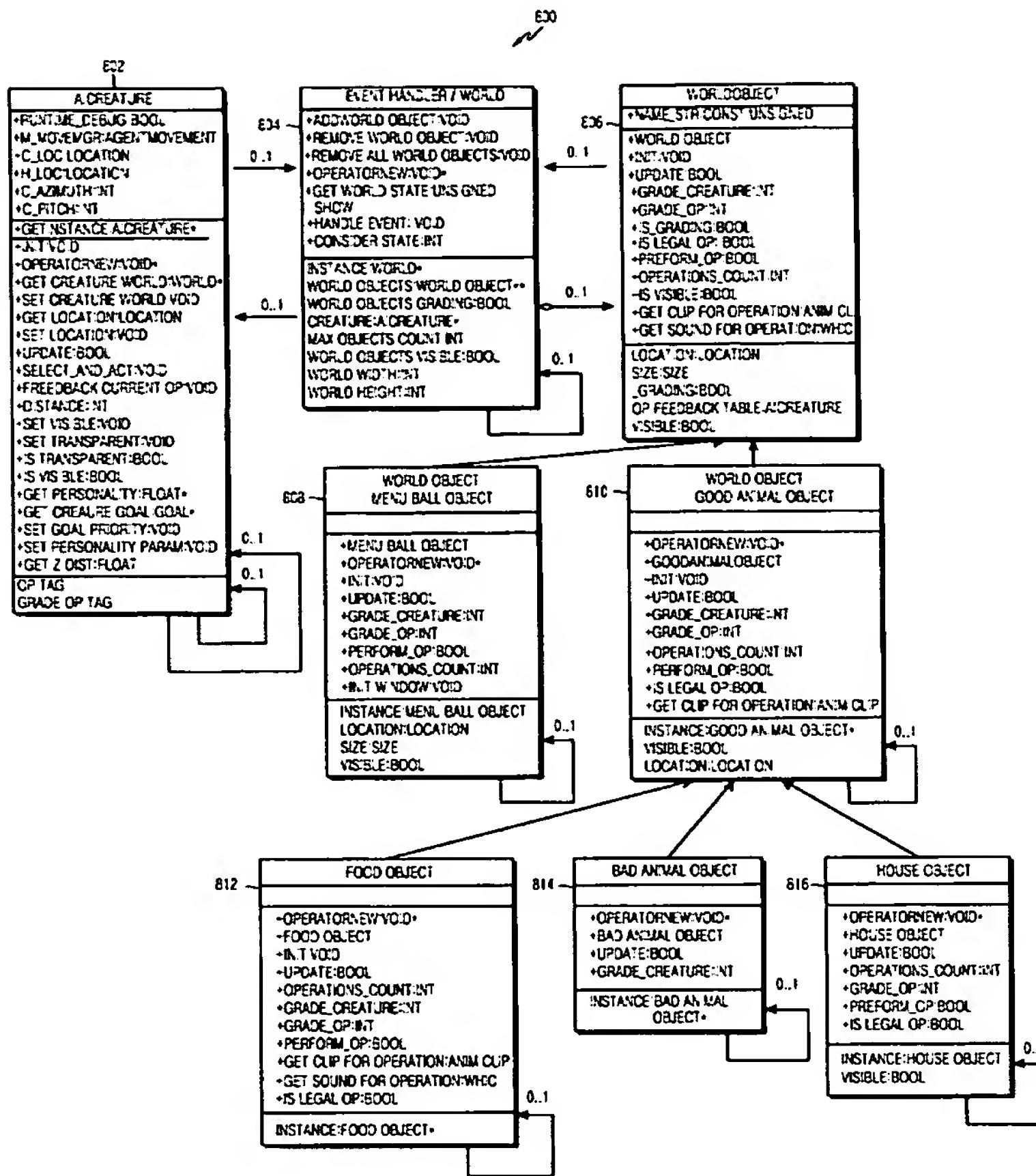
도면 7b



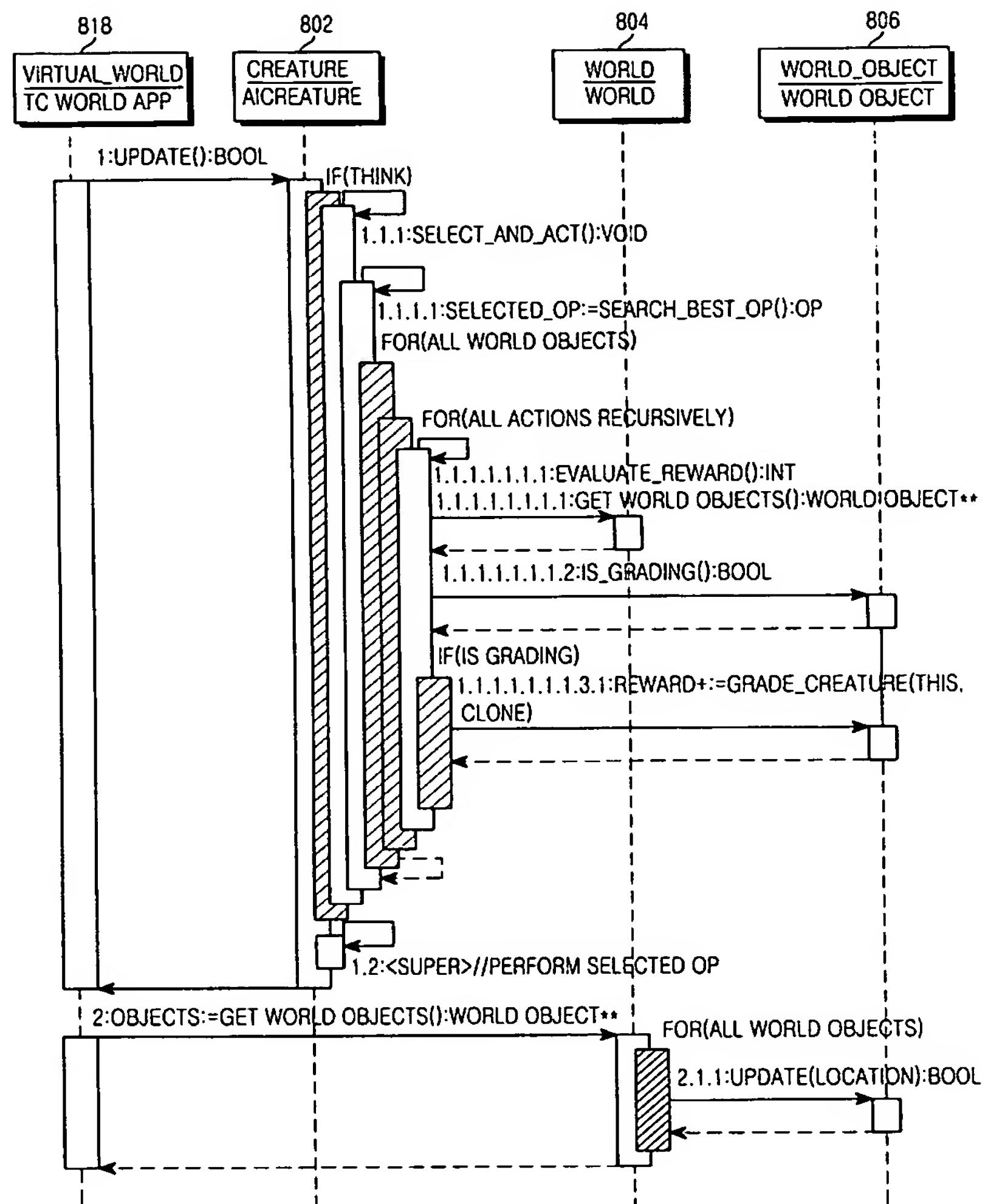
도면 7c



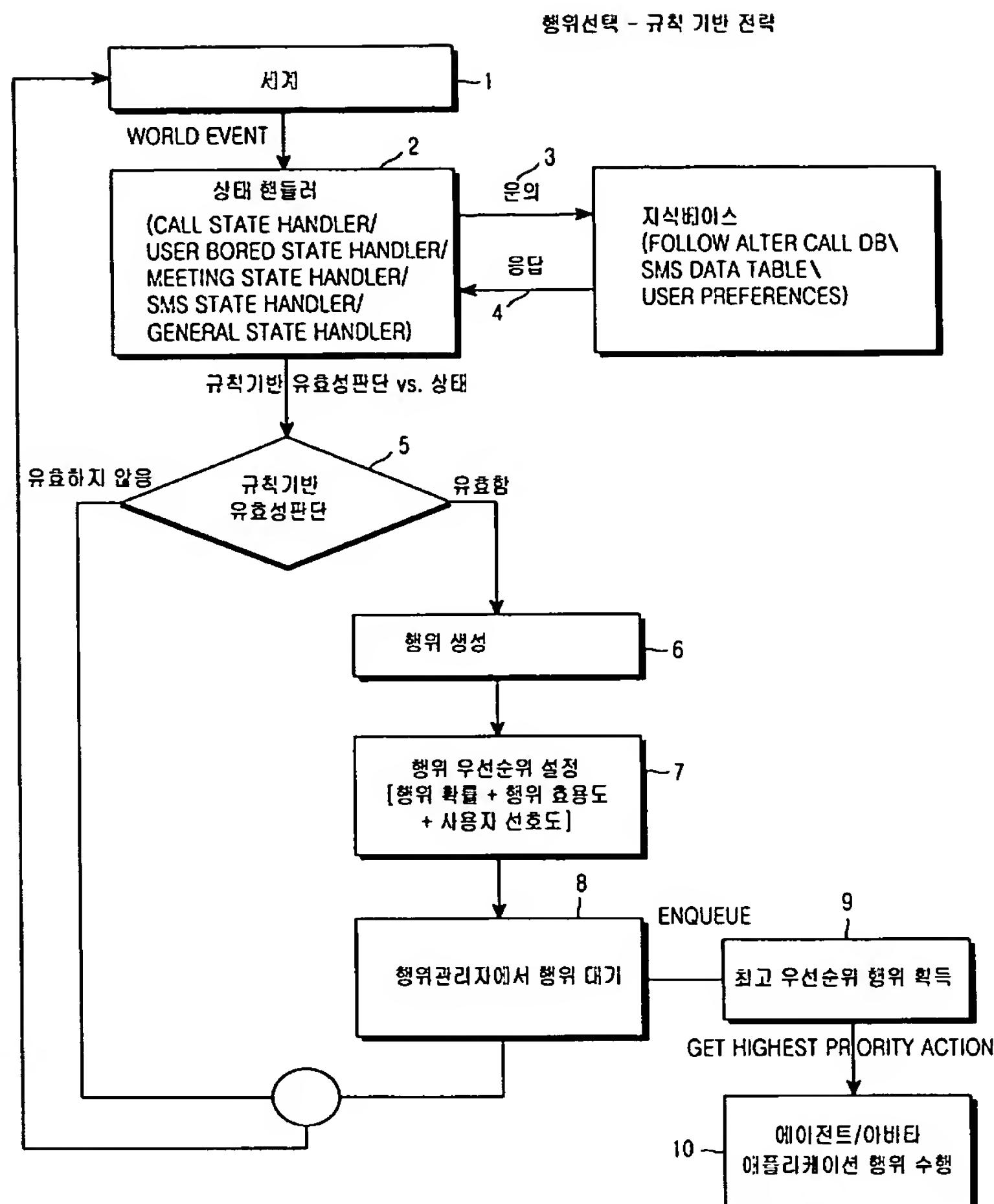
## 도면 8a



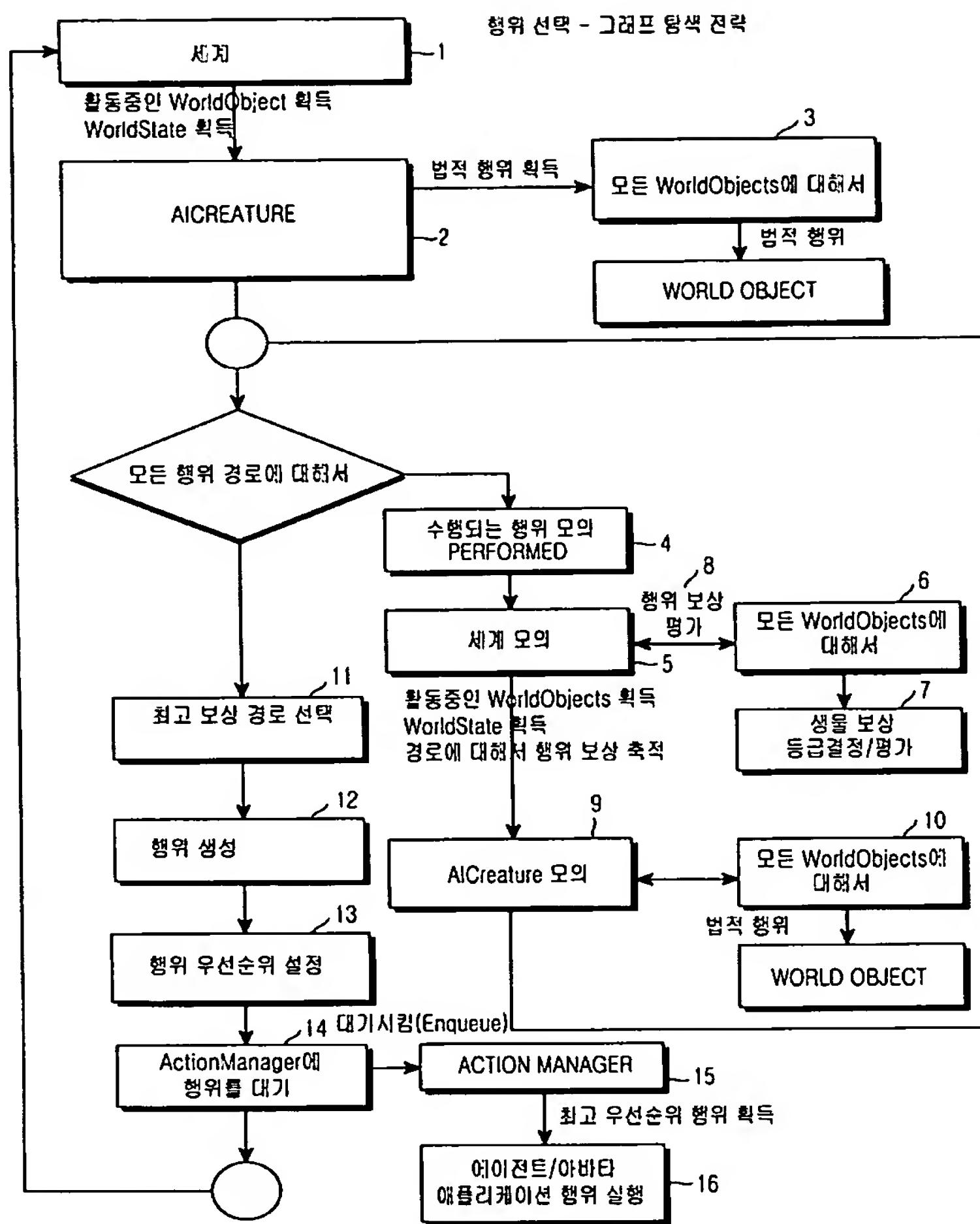
도면 8b



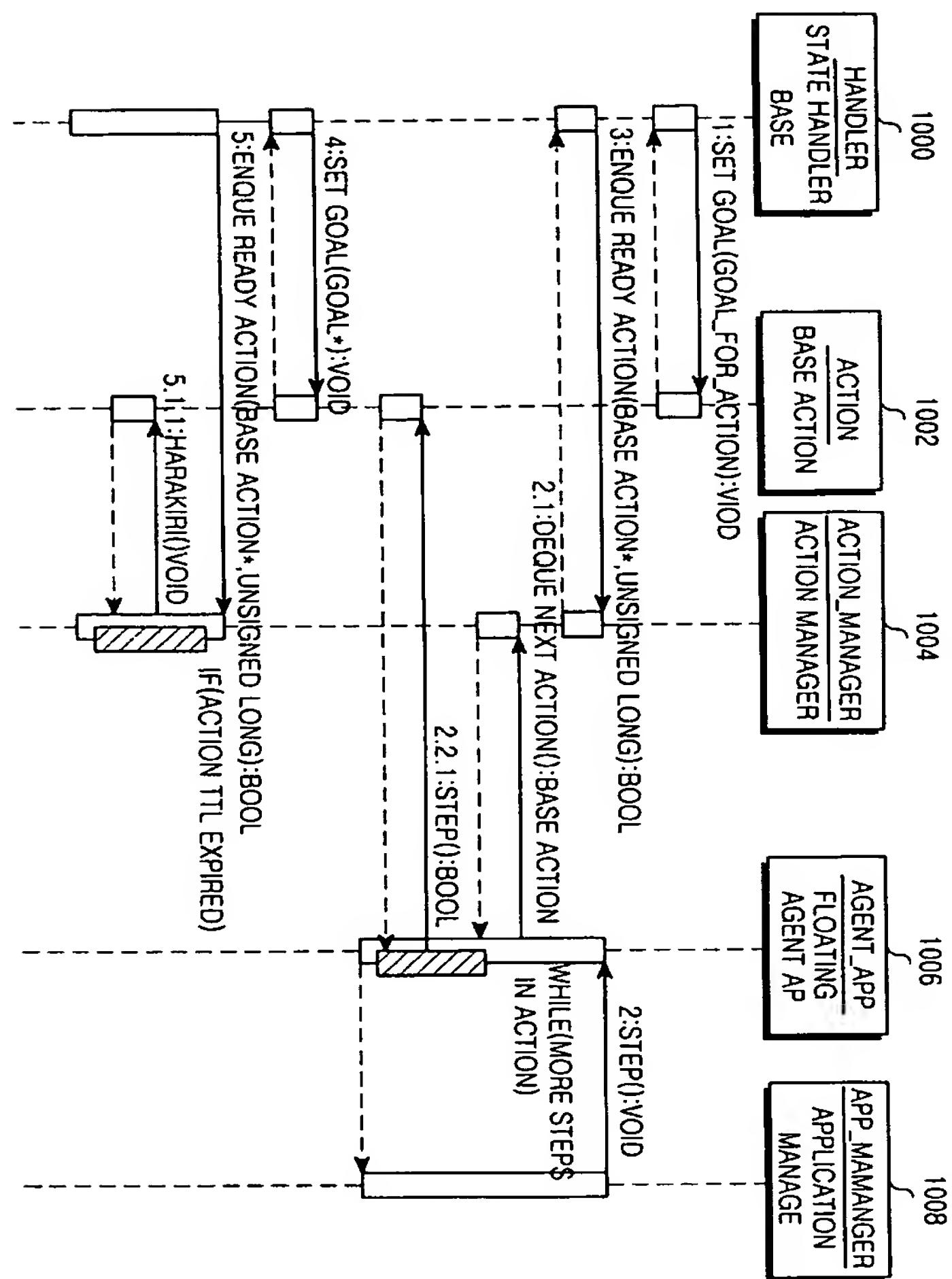
도면9a



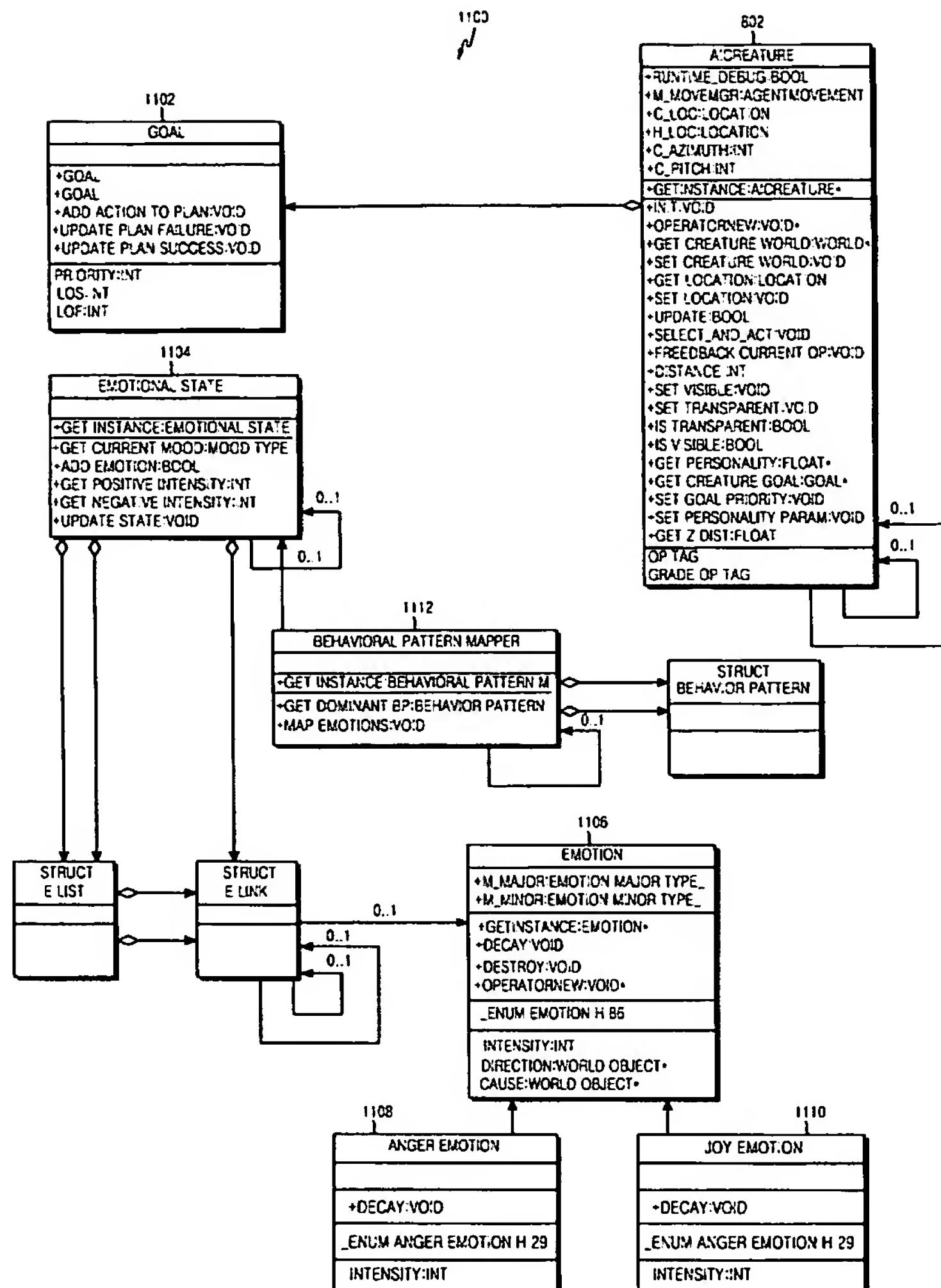
도면 9b



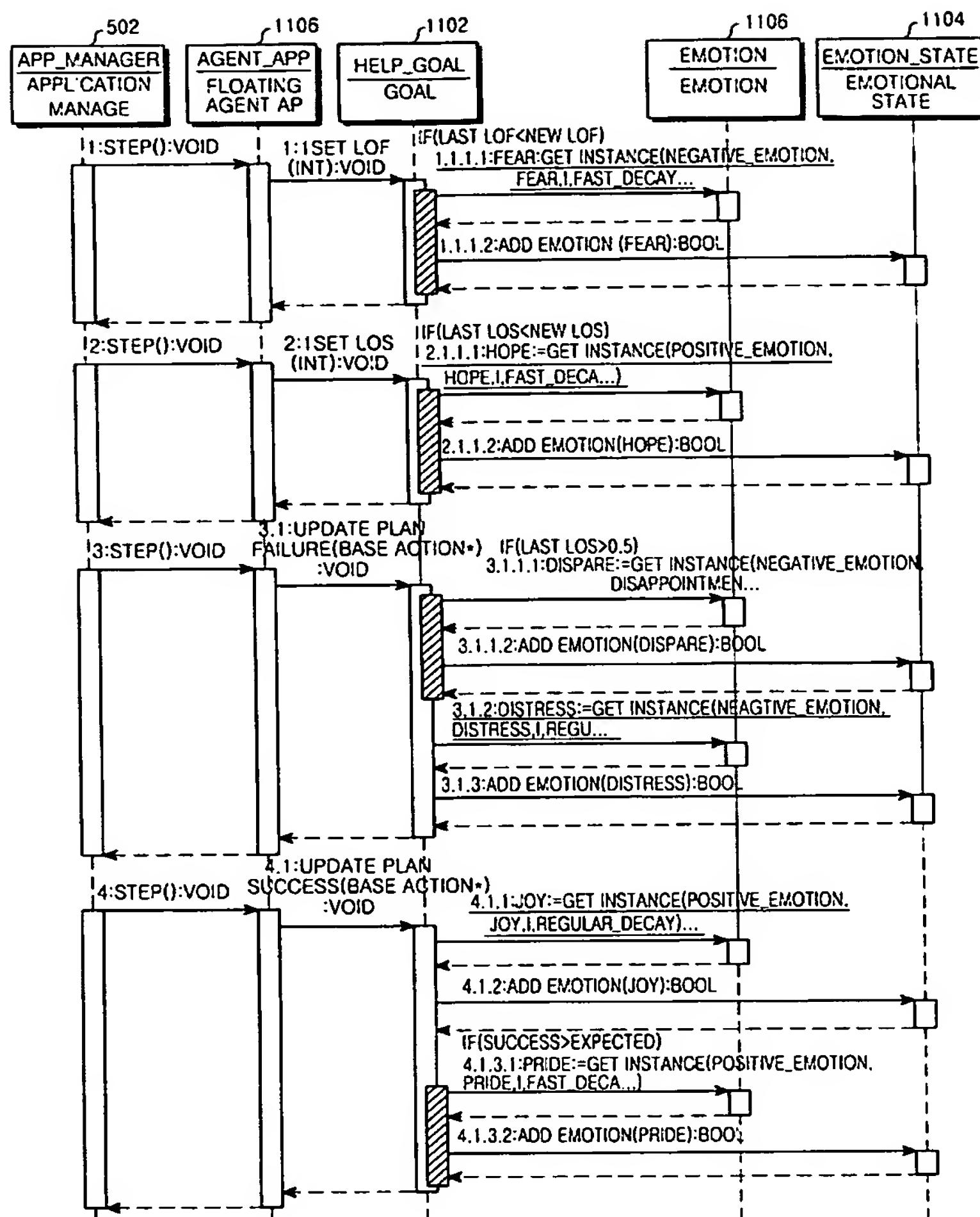
도면 10



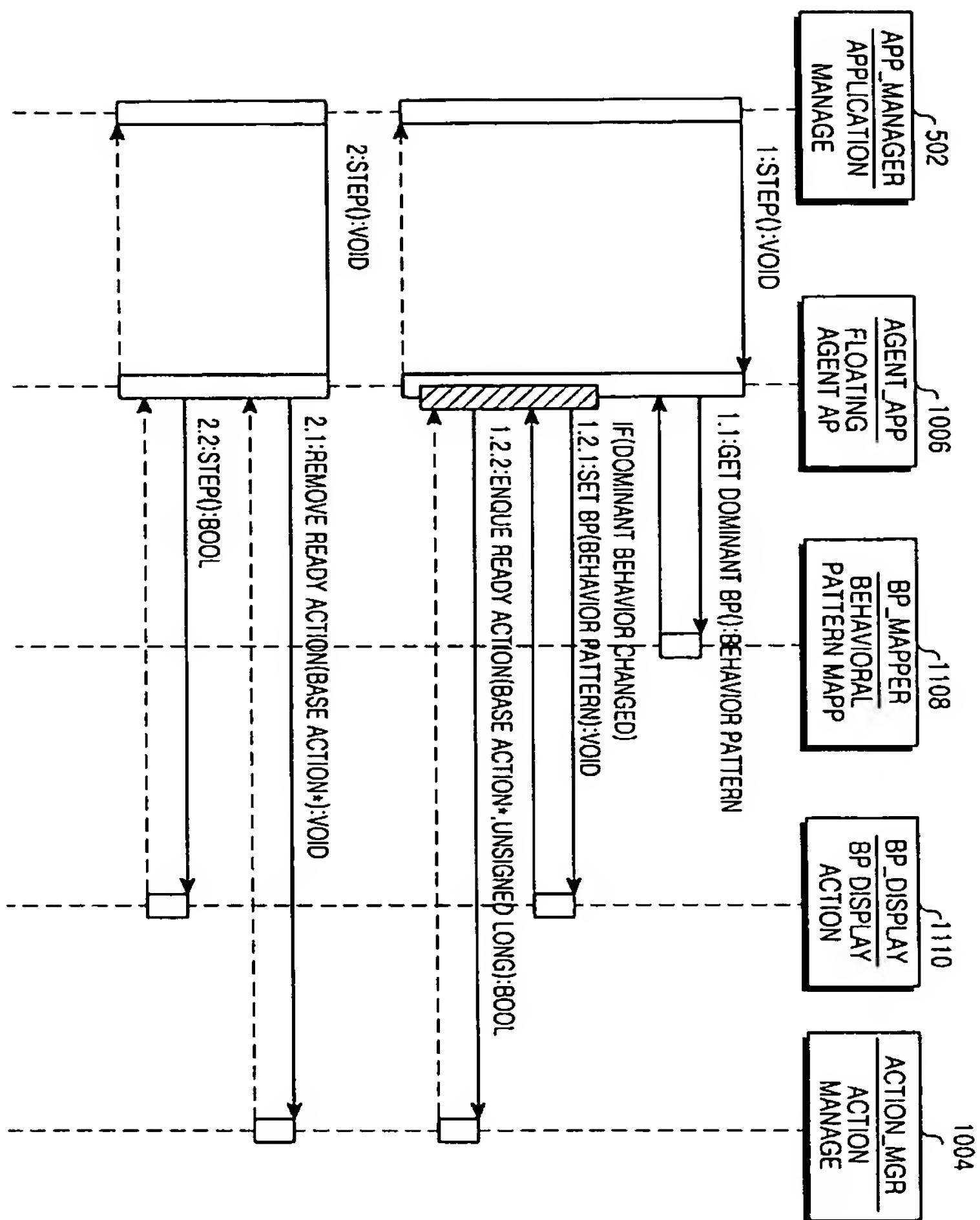
도면 11a



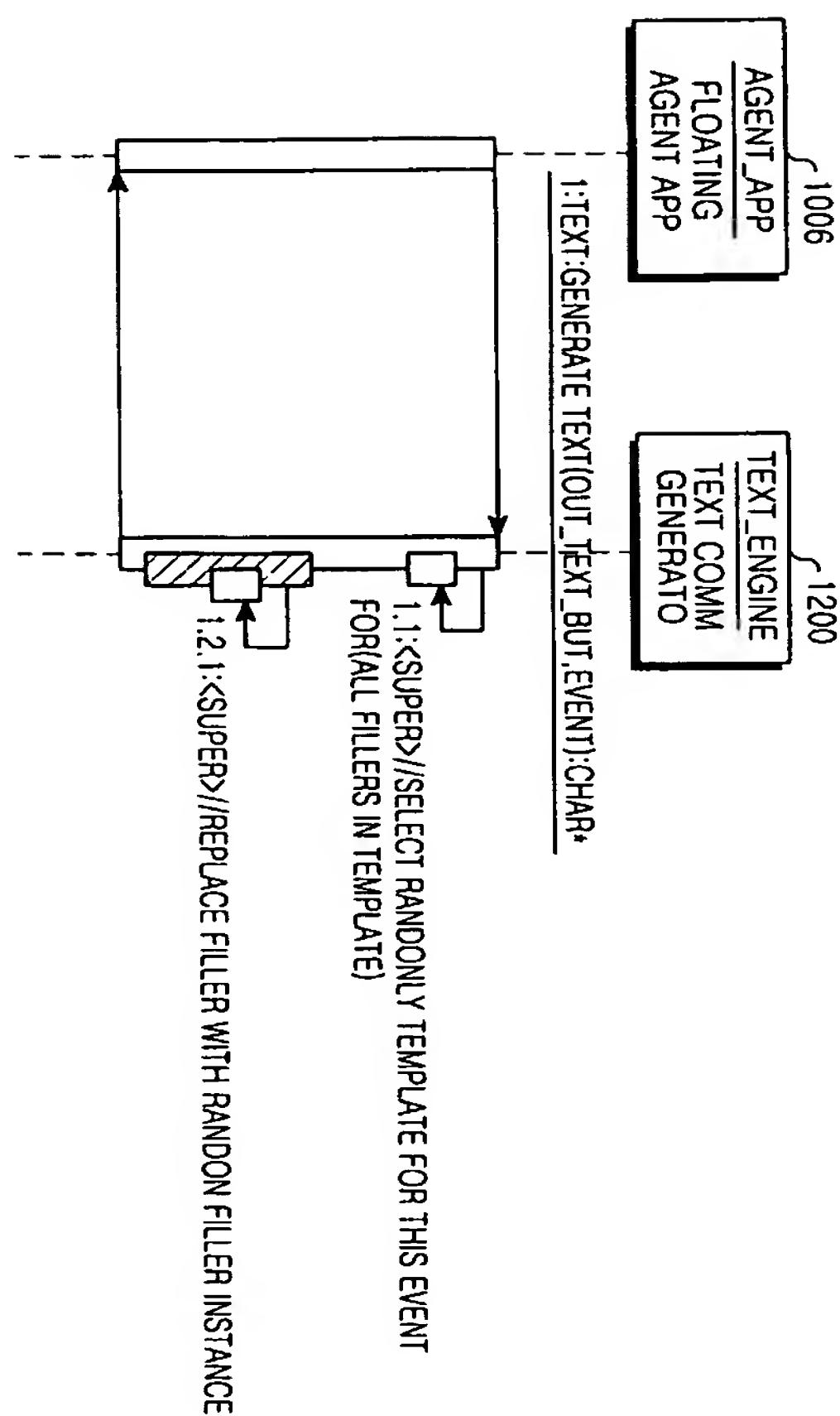
## 도면 11b



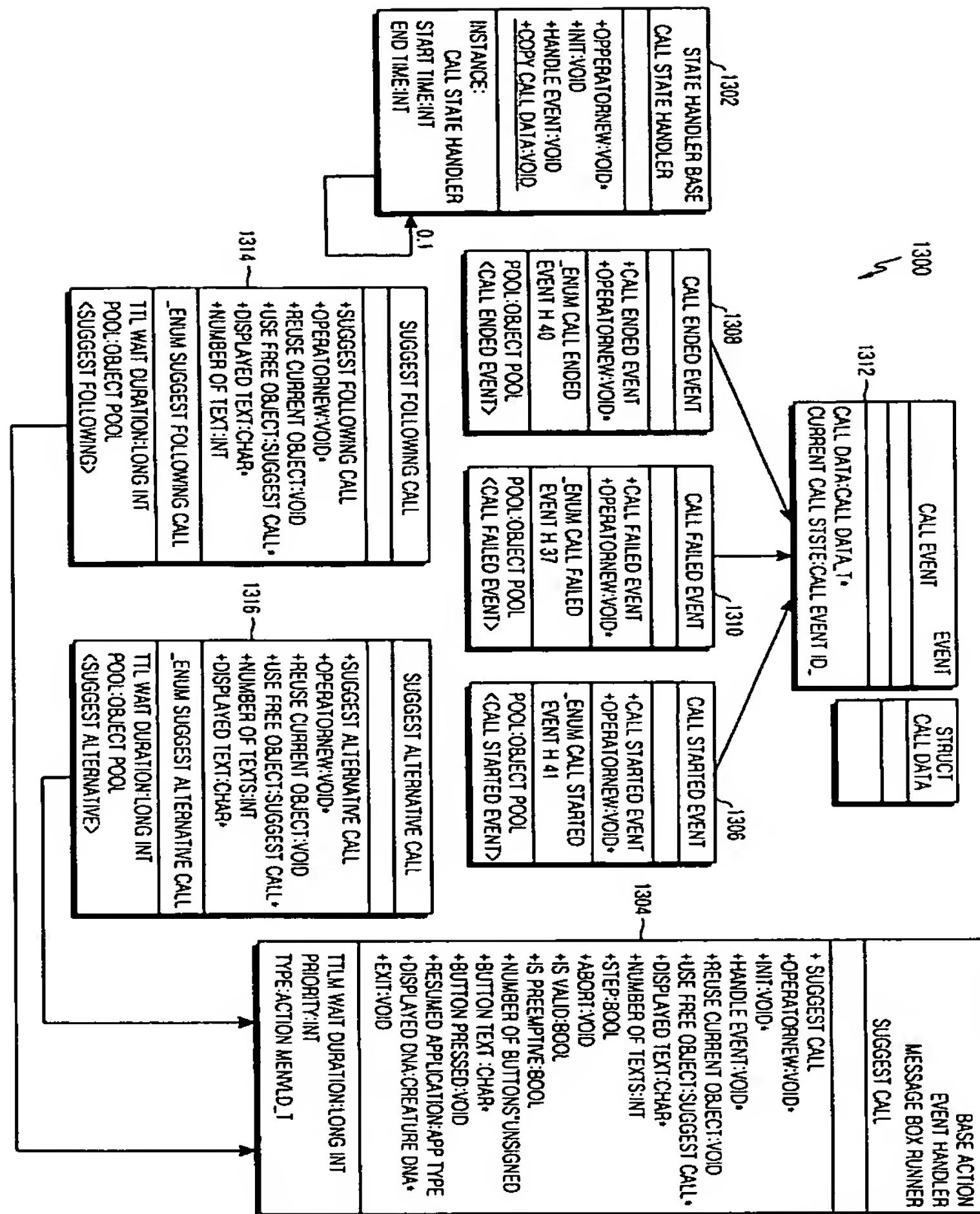
도면 11c



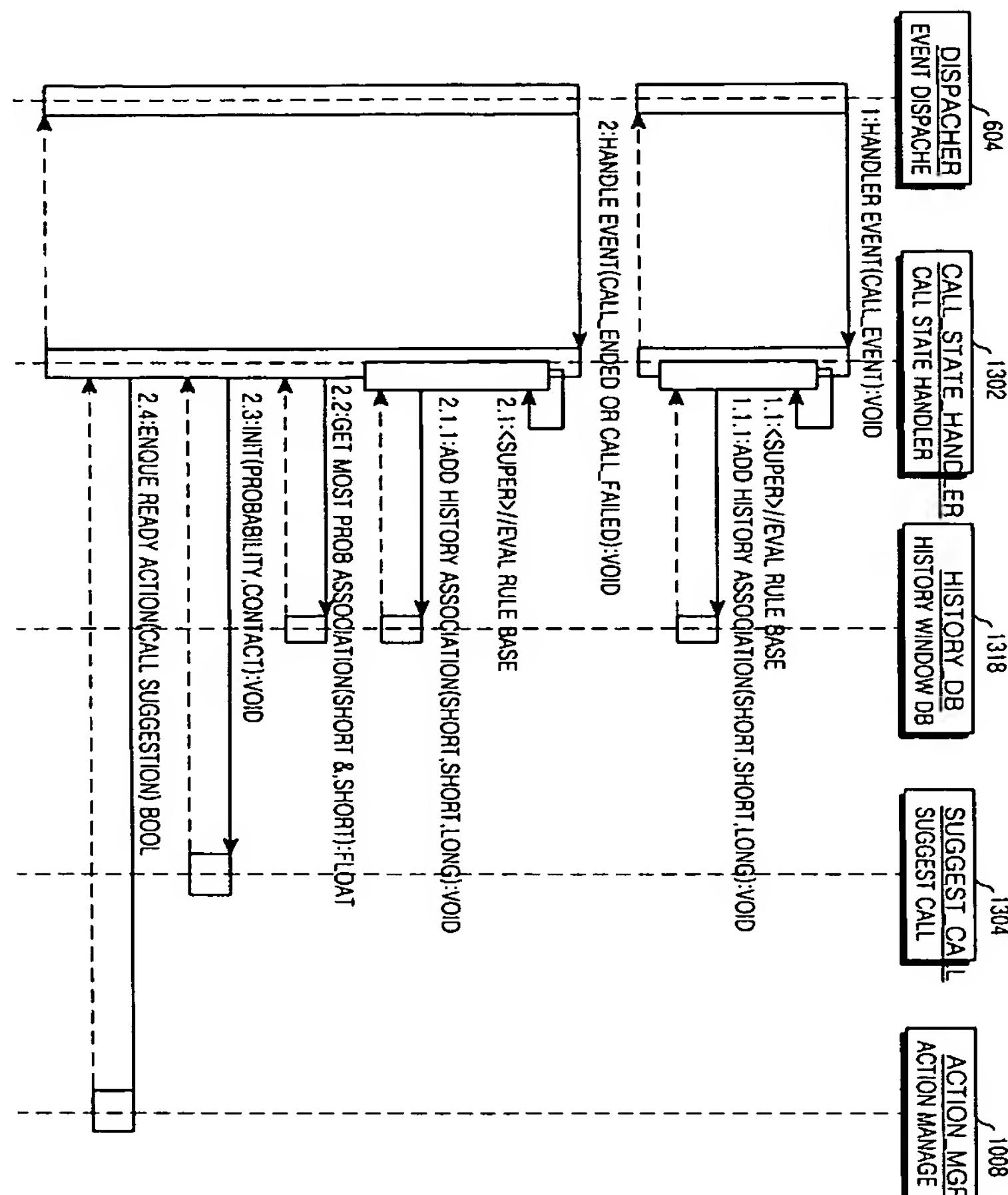
도면 12



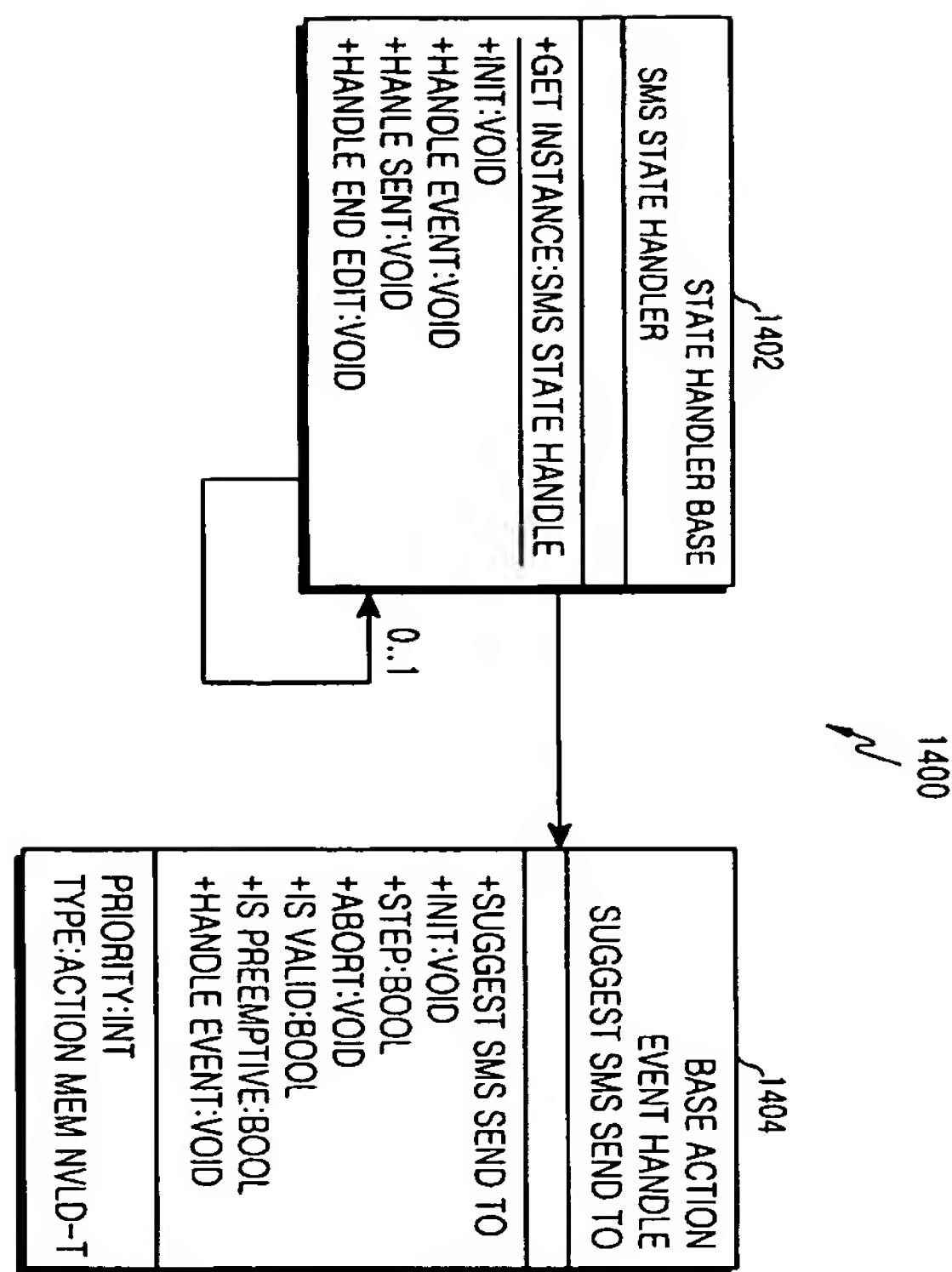
### 도면 13a



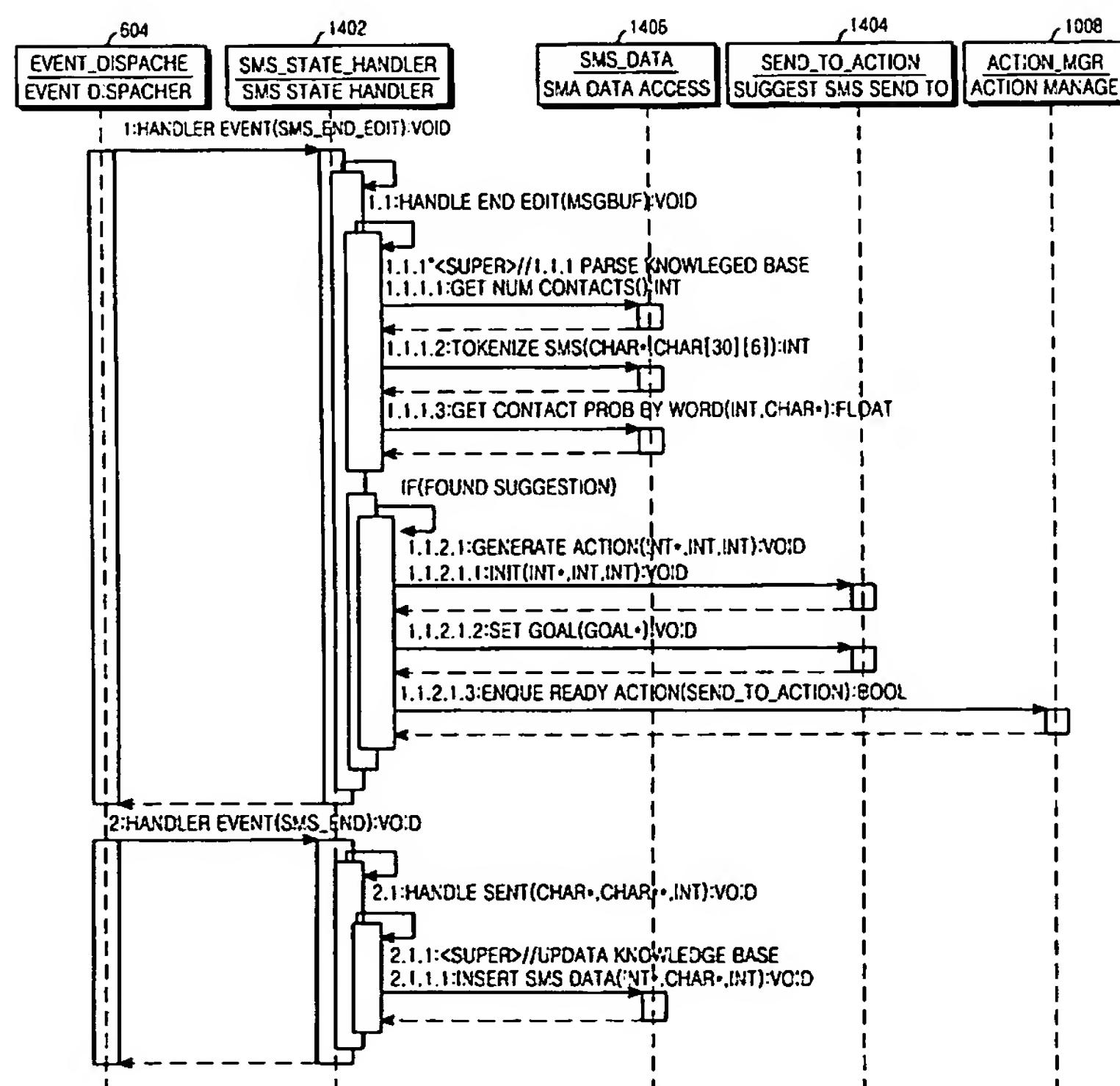
도면 13b



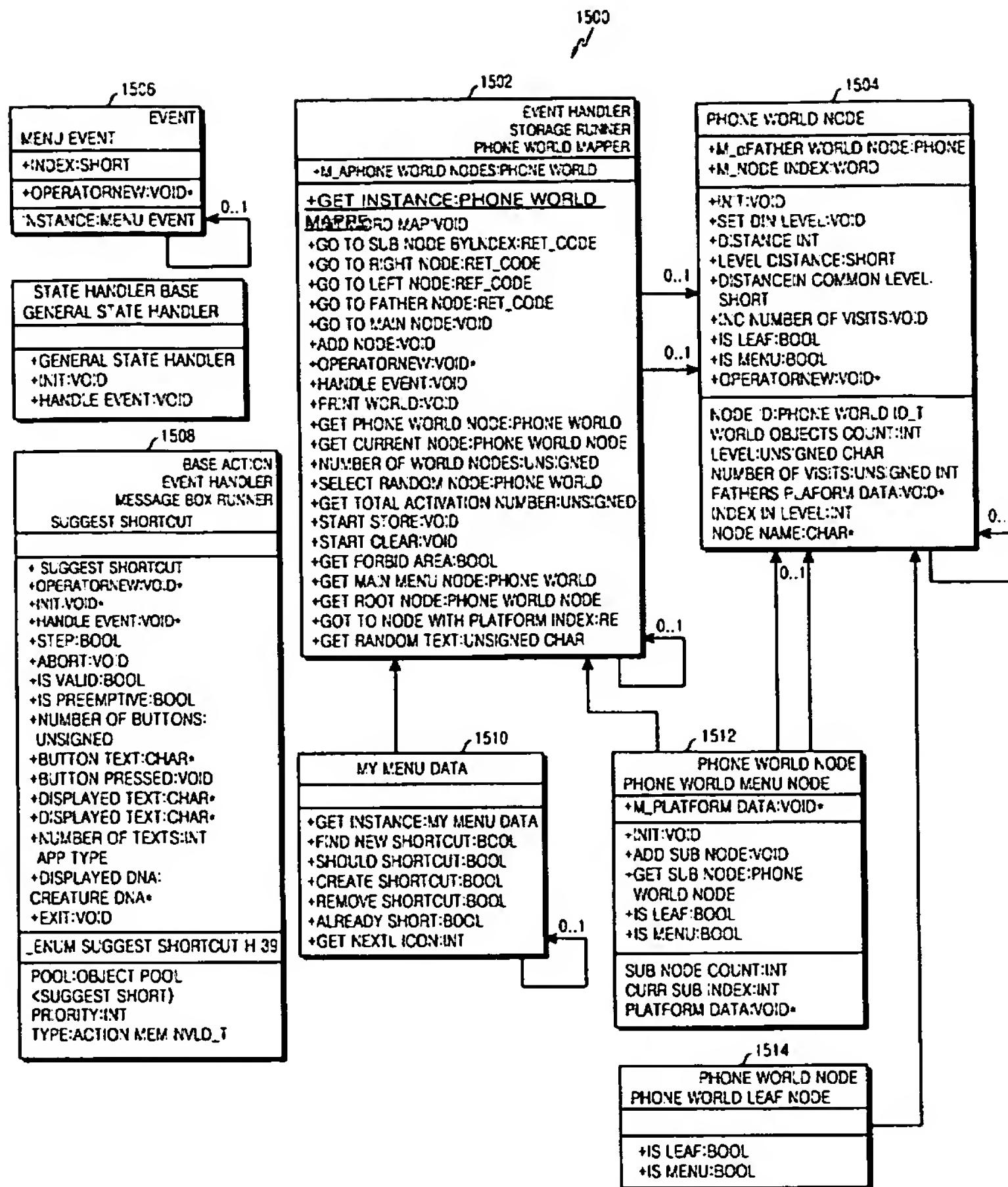
도면 14a



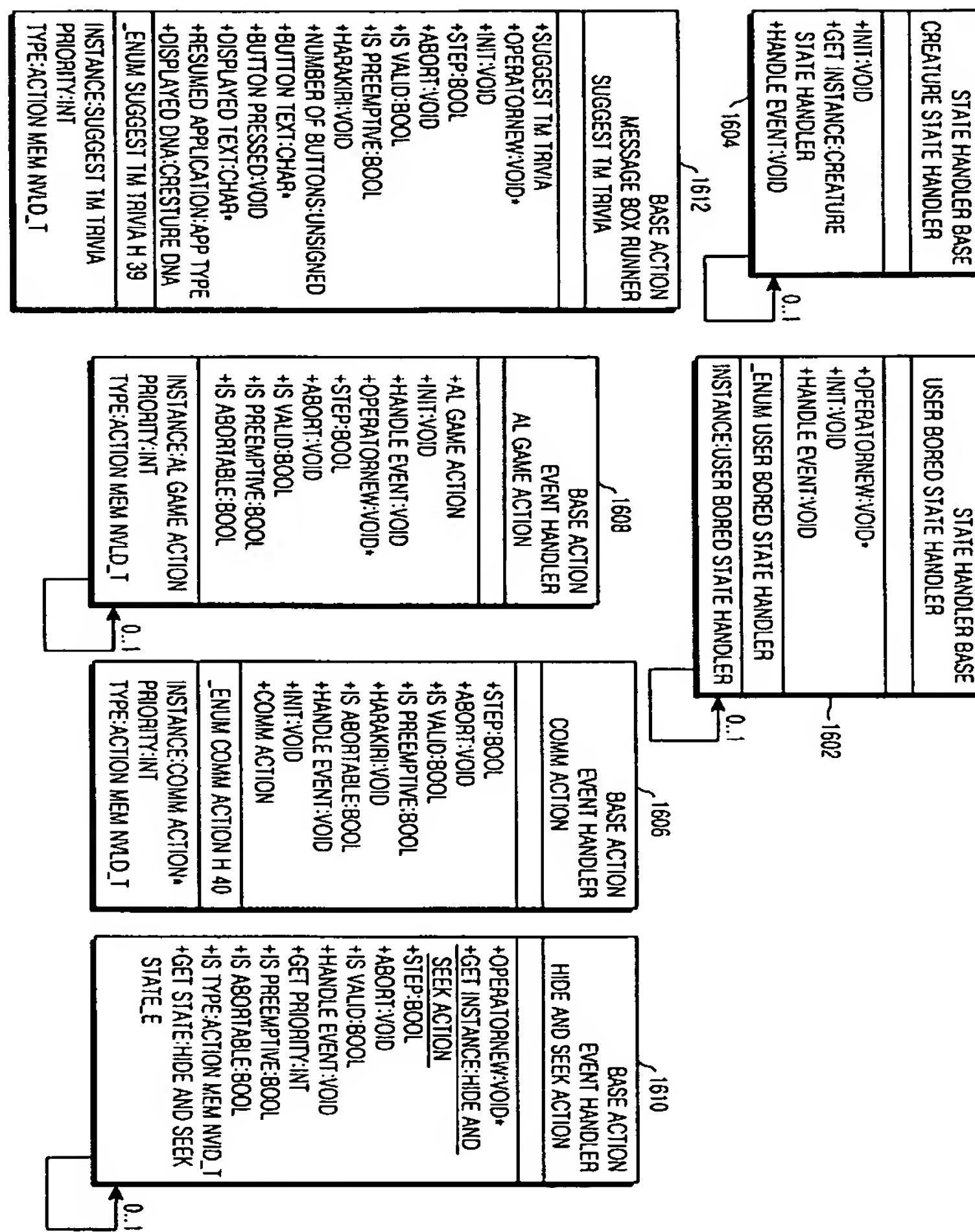
도면 14b



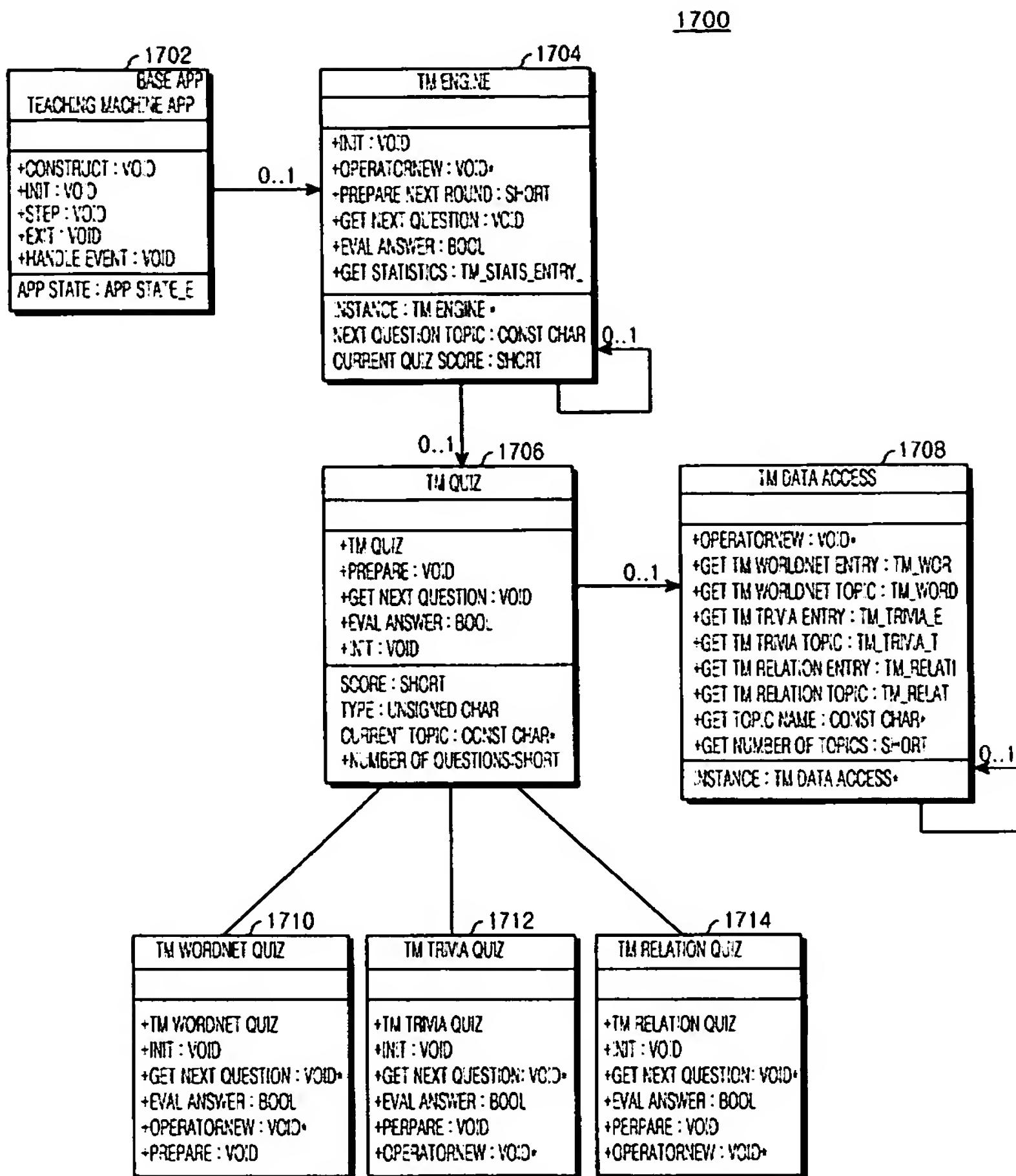
## 도면 15



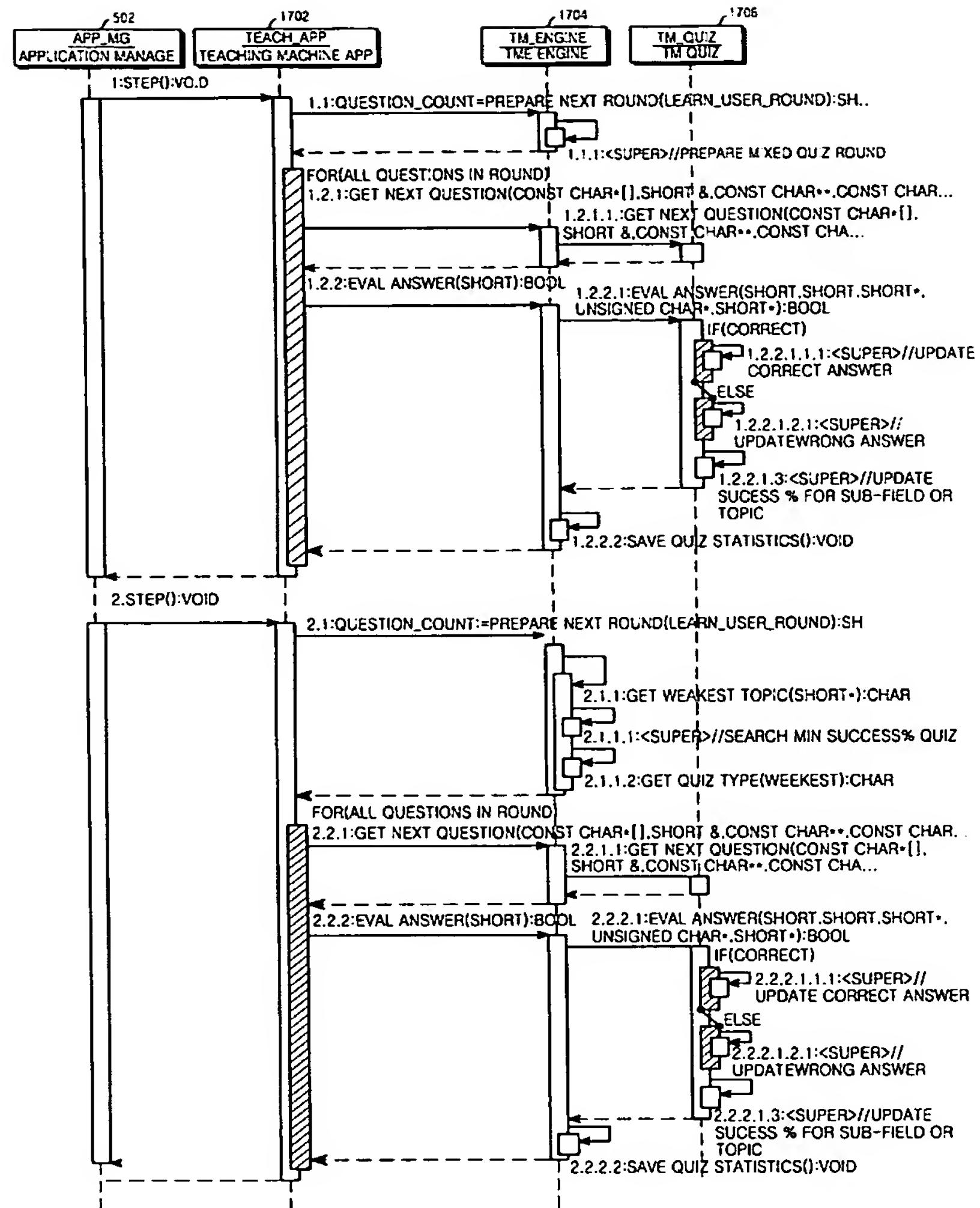
## 도면 16



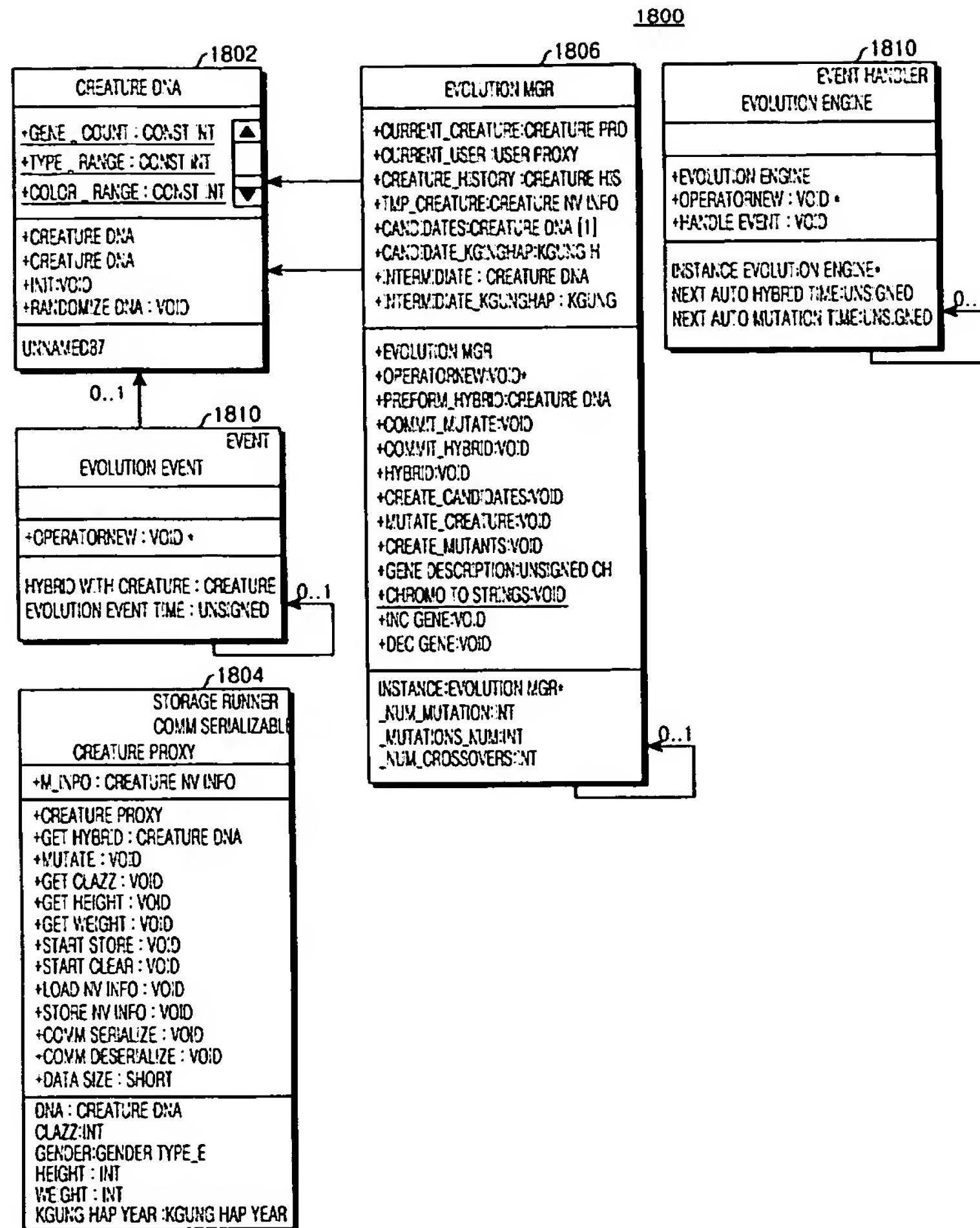
도면 17a



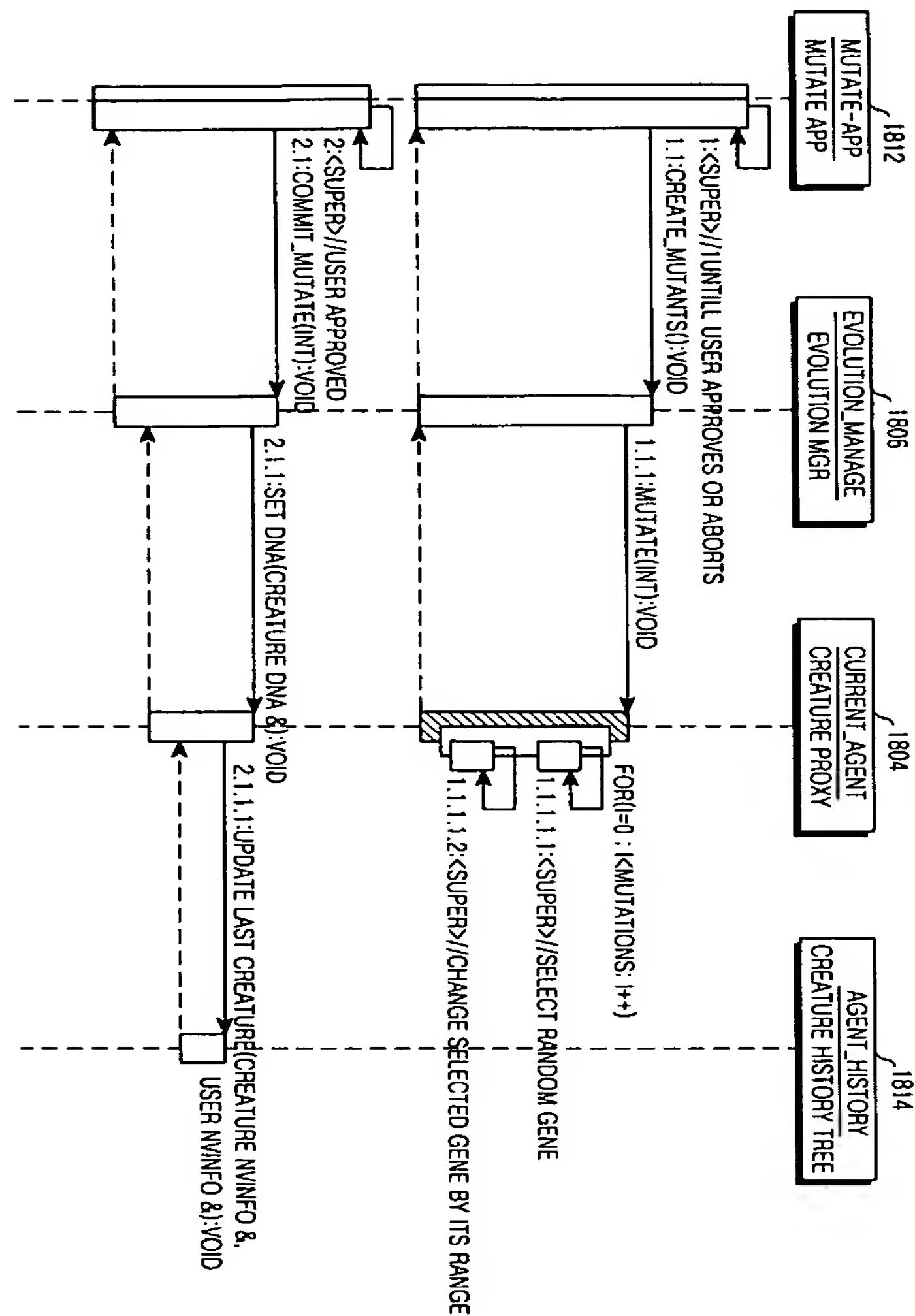
## 도면 17b



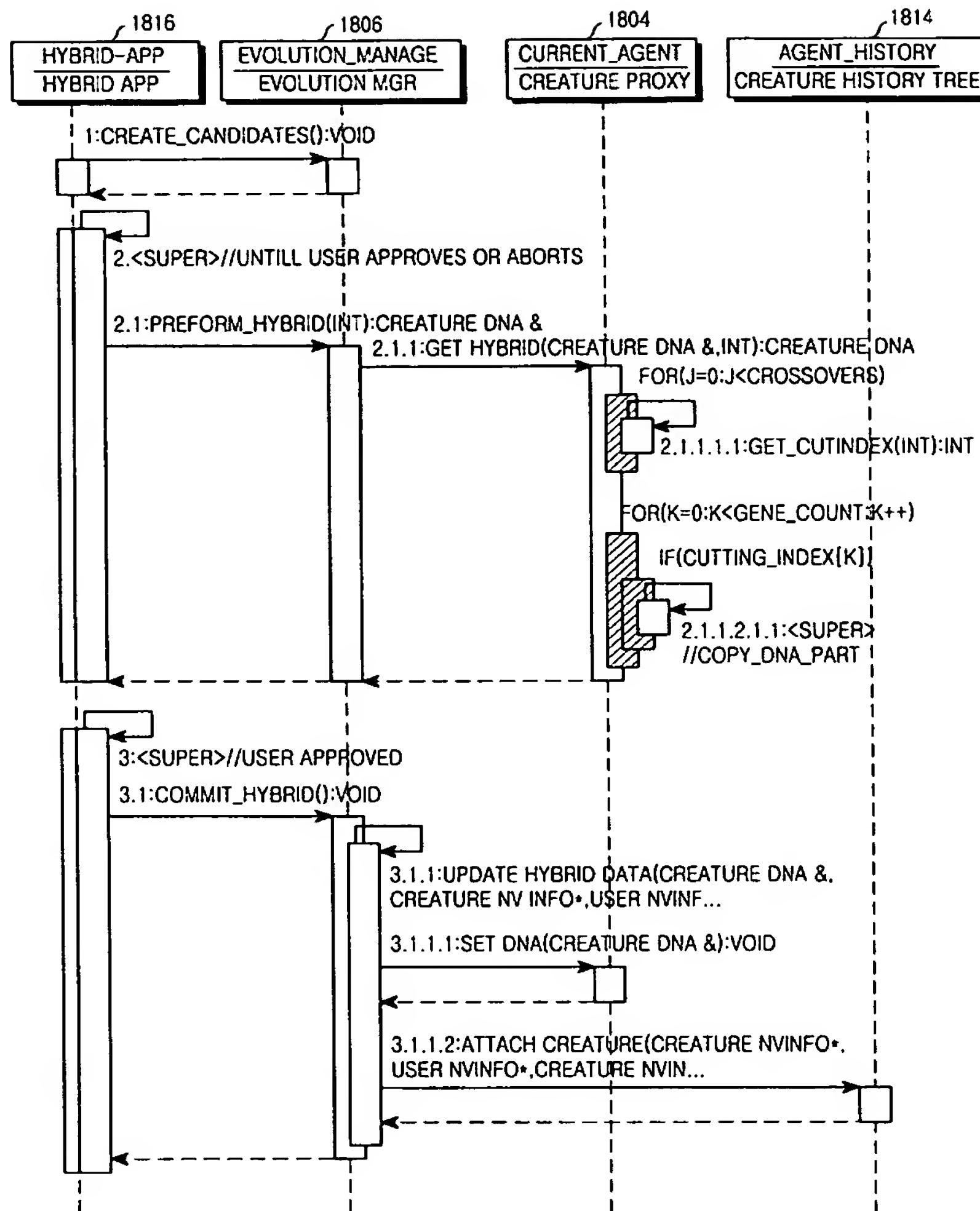
도면 18a



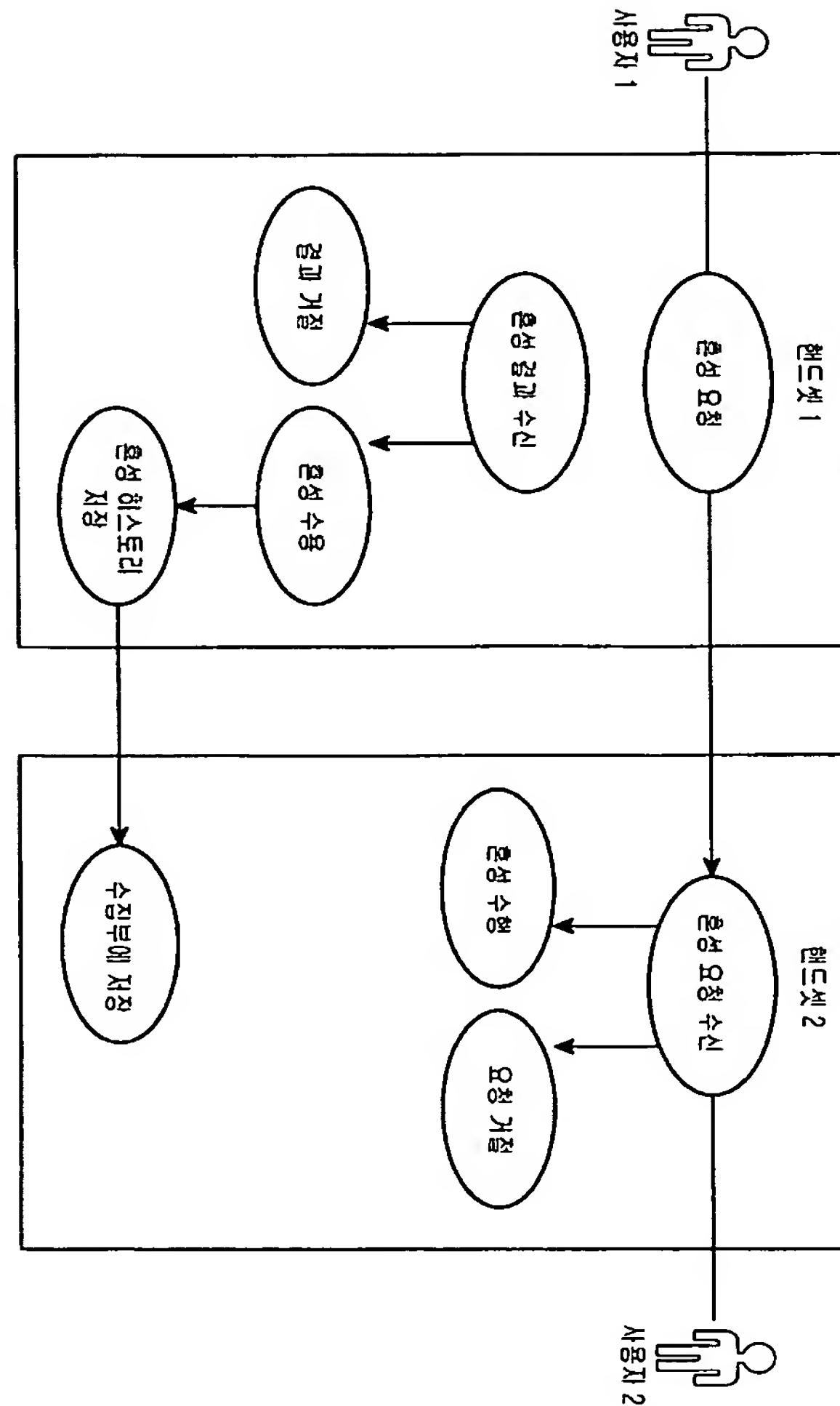
도면 18b



도면 18c



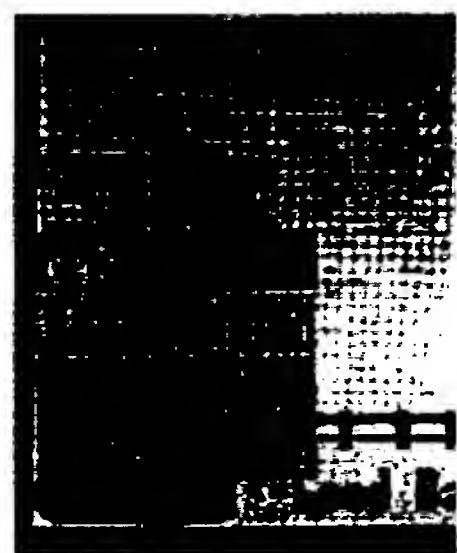
도면19



도면20



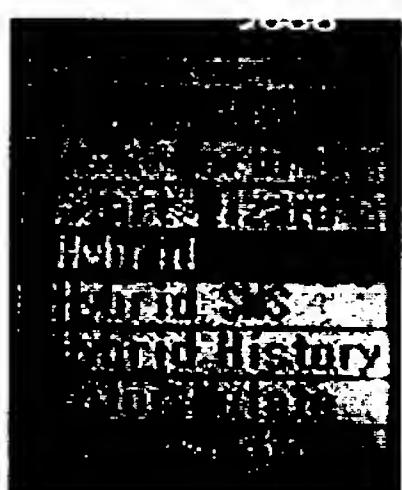
도면21



도면22



도면23



도면24a



도면24b



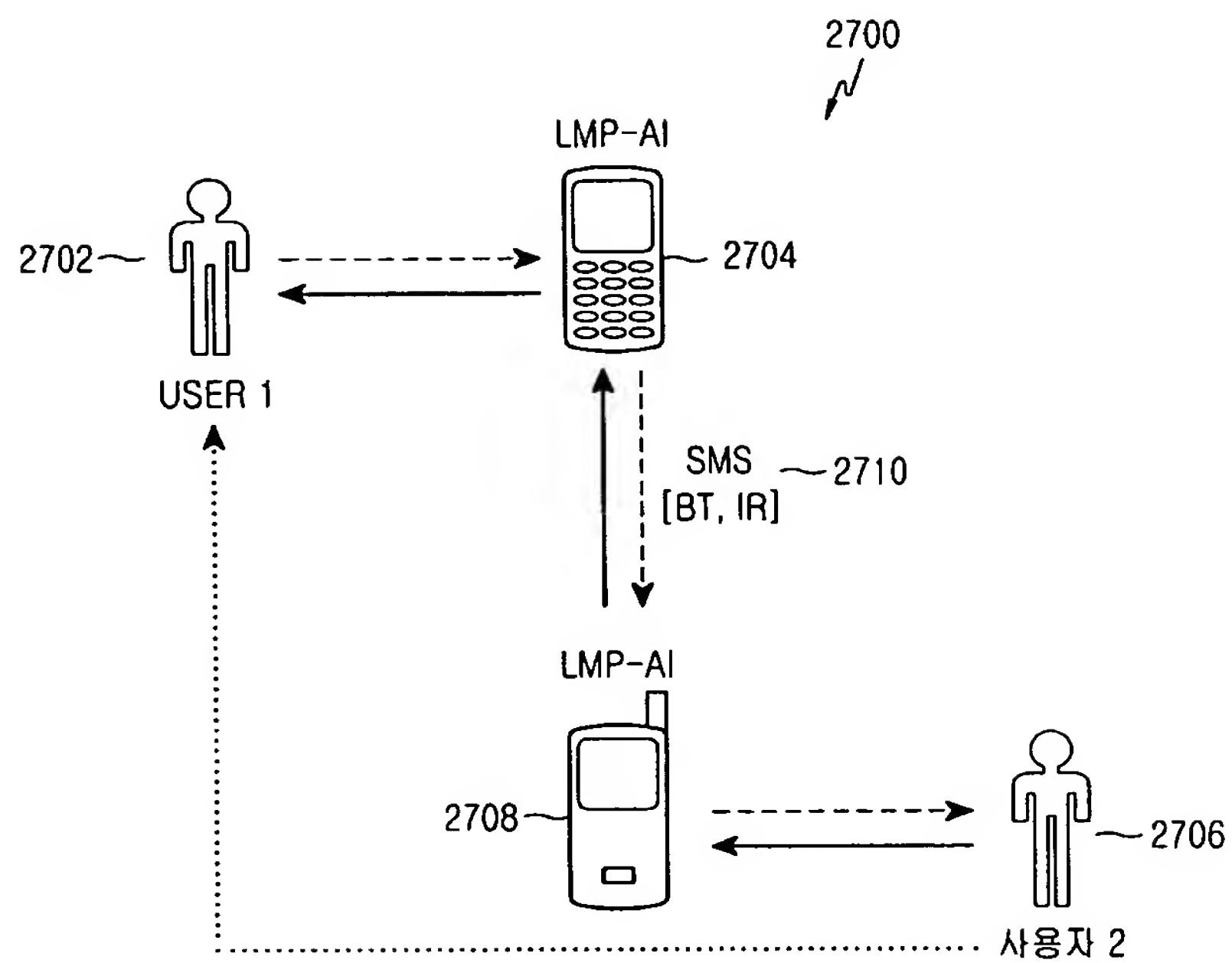
도면25



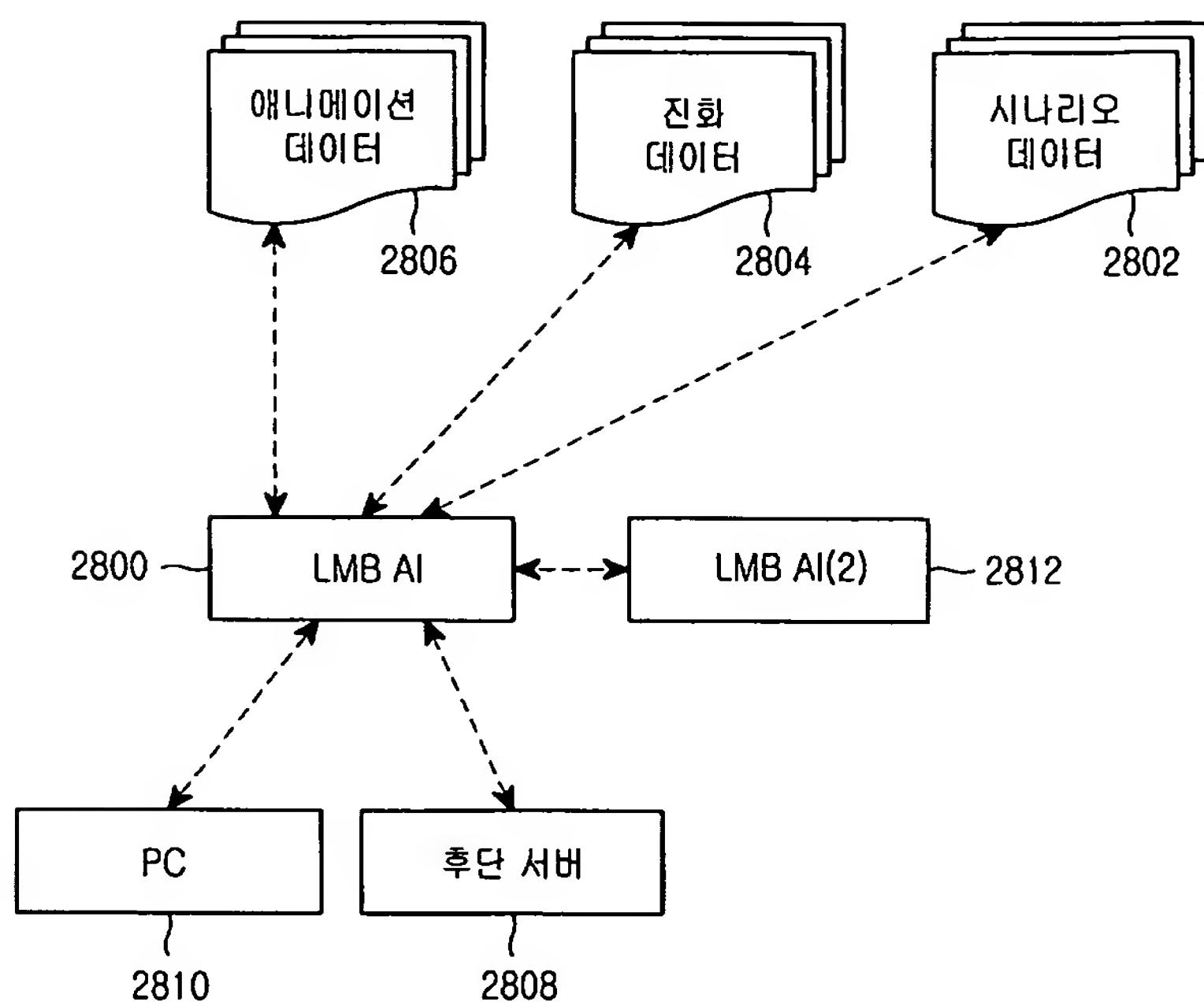
도면26



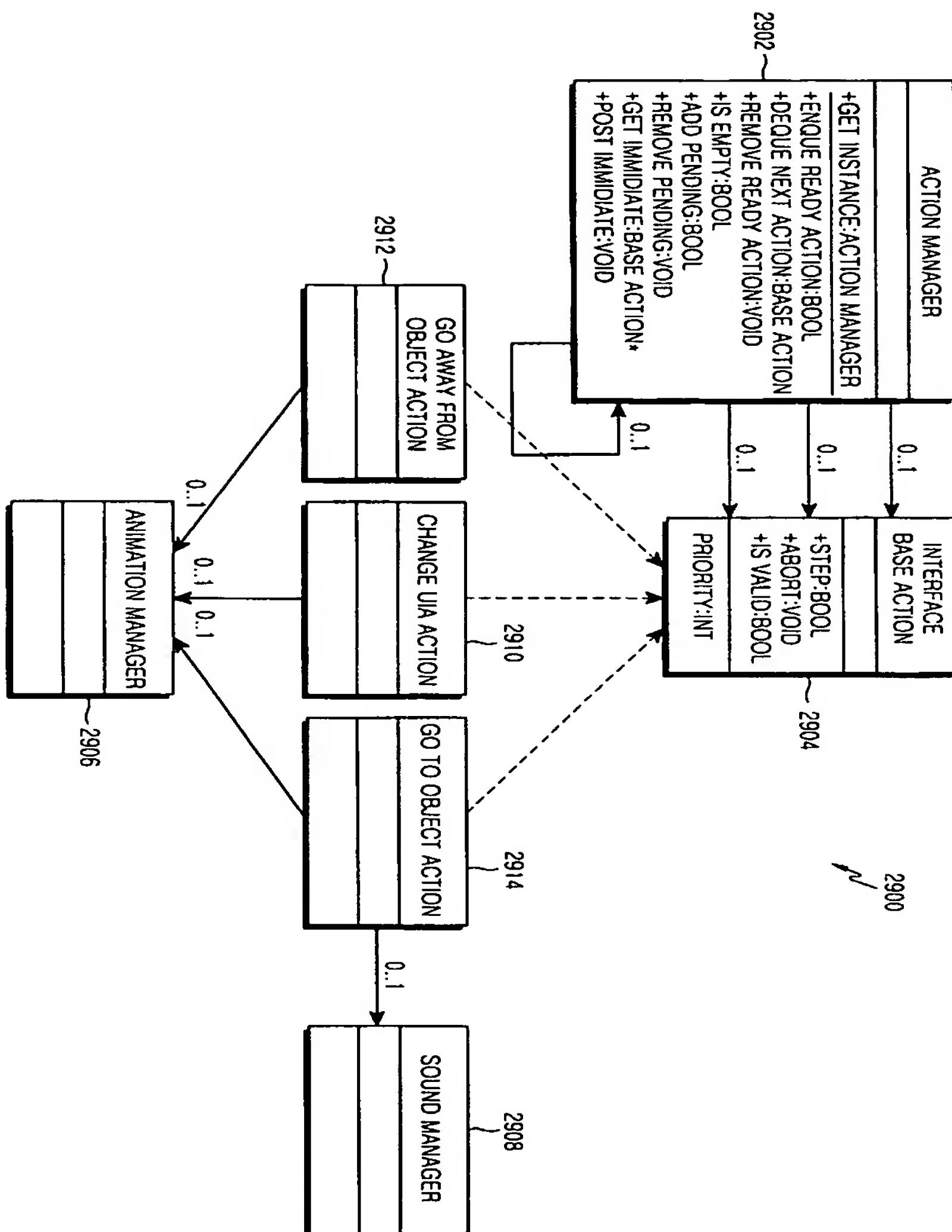
도면27



도면28



도면29



도면30a



도면30b

